Bayesian networks for predicting duration of phones

Olga Goubanova

Thesis submitted for the degree of Doctor of Philosophy

University of Edinburgh

June 2005

Dedication

To my parents

Acknowledgements

I would like to give my thanks to the people who knowingly or unknowingly helped me to have this thesis written. Many thanks goes to my family, primarily to my mom Ira, my mother-in-law Sveta, my husband Rodion and my son Andrey: you were helpful, inspiring, critical, loud and funny. And thanks to my late father, who taught me to achieve.

I would like to thank my many friends in Russia, Spain, Zimbabwe, the United States and Scotland. Thanks to Elena Minyonok, my dear friend who taught me to dream and to believe. I am thankful to Sergei Minyonok, who taught me to observe and see hidden things. I would like to thank my dearest friends Dee Worman and Karl Reynolds. They taught me to de-construct, analyse and sythesise. I would like to thank Bryant York, who taught me not to give up and look for economical and pratical solutions.

I am very thankful to my linguist friends, former or current Ph.D. students from the Department of Linguistics, now the Department of Theoretical and Applied Linguistics: Marisa Flecha-Garcia, Helena Alfaya Llamas, Ivan Yuen, Kook-hee Gil, George Tsoulas, Patricia Mabugu, Karen Sode-Woodhead. They were great listeners, who were always ready to share a fiver or a laughter.

I would like to thank my co-workers, present and former colleagues from the Centre for Speech Technology Research: Paul Taylor, John MacKenna, Sue Fitt, Joe Frankel, Korin Richmond, Rob Clark, Alex Gukin. They were always ready to help, be it a minute problem or a time-consuming task. I would like to thank Steve Isard and Bob Ladd, Alice Turk and Ellen Bard,

Robin Lickley, Jim Hurford and Jim Miller who were always helful and ready to share thoughts, information and what not. I would like to thank my supervisor Dr. Simon King, who guided me through many obstacles. Being a great teacher, he was attentive, punctual, inspiring and firm enough to have me think and experiment which finally resulted in my thesis written.

Declaration

I have composed this thesis. The work reported in this thesis is my own, unless otherwise mentioned.

Abstract

In a concatenative text-to-speech (TTS) system, the duration of a phonetic segment (phone) is predicted by a duration model which is usually trained using a database of feature vectors, that consist of a set of linguistic factors' (attributes') values describing a phone in a particular context. In general, databases used to train phone duration models are unbalanced. However, it has been shown that the probability of a rare feature vector occurring even in a small sample of text is quite high. Furthermore, factors affecting phone's duration interact; a set of two or more factors may amplify or attenuate the affect of other factors. A robust model for predicting phone duration must generalise well in order to successfully predict the durations of phones with these rare feature vectors. Since linguistic factors affecting segment duration interact, we would expect that modelling these factor interactions will give a better model. There have been a number of models developed for predicting a phone's duration, ranging from rule-based to neural nets to classification and regression tree (CART) to sums-of-products (SoP) models. In the CART model, a phone's duration is predicted by a decision tree. The tree is built by recursively clustering the training data into subsets that share common values for certain attributes of the feature vectors. The duration of a phone is then predicted by using the tree to find the data cluster that matches as many of the feature vector attributes as possible. The CART model is easy to build, robust to errors in data but performs poorly when the percent of missing data is too high. In the SoP model, the log of a phone's duration is predicted as a sum of factors' product terms. The SoP model predicts phone duration with high accuracy, even in cases of hidden or missing data. However, this is done at the cost of substantial data pre-processing. In addition, the number of different sums-of-products models grows hyper-exponentially with the number of factors. Therefore, one must use some heuristic search techniques to find the model that fits the data the best. In our work, we use a Bayesian belief network (BN)

consisting of discrete nodes for the linguistic factors and a single continuous node for the phone's duration. Interactions between factors are represented as conditional dependency relations in this graphical model. During training, the parameters of the belief network are learned via the Expectation Maximisation (EM) algorithm. The duration of each phone in the test set is then predicted via Bayesian inference: given the parameters of the belief network, we calculate the probability of a phone taking on a particular duration given the observations of the linguistic variables. The duration value with the maximum probability is chosen as the phone's duration. We contrasted the results of the belief network model with those of the sums of products and CART models. We trained and tested all three models on the same data. In terms of the RMS error our BN model performs better than both CART and SoP models. In terms of the correlation coefficient, our BN model performs better than SoP model, and no worse than CART model. We believe our Bayesian model has many advantages compared to CART and SoP models. For instance, it captures the factors' interactions in a concise way by causal relationships among the variables in the graphical model. The Bayesian model also makes robust predictions of phone duration in cases of missing or hidden data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Text-to-speech (TTS) synthesis today

1.2 Duration modelling for text-to-speech synthesis

Speech synthesis is a technology that surrounds us everywhere. The areas of application are too many to list. It is used in telephone directory assistance and call centres query systems. Speech generation is a part of language learning applications. Synthesised speech is part of systems to aid handicapped persons.

Timing is undoubtedly one of the main characteristics of the spoken language that plays an important role in encoding and decoding of the spoken message. Timing is equally important for synthesised speech: one of the main requirements of any synthesiser is to sound natural and intelligible in order to be understood and accepted by listeners. Consequently, the need arises for a robust duration model as a part of a high quality text-to-speech (TTS) system.

A text-to-speech synthesiser is a complex computer based system for reading a text aloud. It takes as an input a plain text transcription of the sentences and produces as an output a speech waveform. In a concatenative

text-to-speech (TTS) system, the duration of a phonetic segment is predicted by a duration model. A duration model is trained on a database of feature vectors, that consist of a set of linguistic factors' (attributes') values describing a phone in a particular context. In general, databases used to train phone duration models are unbalanced. The linguistic space occupies an uneven part of the whole feature space[1]. Furthermore, the feature vectors of the training database cover only a fraction of the linguistic space. However, as was shown by van Santen (1994), the joint probability mass of all rare feature vectors taken together is quite large.

In addition, there exists a problem of factor confounding: different feature vectors occur with unequal frequencies in the training database. As a result, *raw* durations calculated from the database can be deceptive. van Santen (1994) gives an example of within-word position and stress factor confounding. Durations of vowels turn out to be shorter in word-final syllables than in non word-final syllables, if stressed and unstressed vowels are analysed together. Given that unstressed vowels are shorter than stressed vowels and the word-final syllable are five times more likely to be unstressed, this gave rise to this result. If stressed and unstressed vowels were analysed separately, the vowel duration in final syllables would be longer than in non-final syllables, as it should be.

Furthermore, factors affecting a phone's duration interact; a set of two or more factors may amplify or attenuate the affect of other factors. van Santen (1994) shown that these effects are quite regular.

A robust model for predicting phone duration must address all of these issues. It should generalise well in order to successfully predict duration of a phone with a feature vector that has hidden (missing) feature values. A robust duration model must properly describe factor interactions. It also must account for factor confounding problem. We expect that modelling factor interactions and accounting for factor confounding will give a better

[1] A feature space is defined as *Cartesian* product of all the factors under consideration: $\mathcal{F} = F_1 \times F_2 \ldots \times F_n$.

model.

1.2.1 Phone vs Syllable modelling

There has been an on-going debate of whether to use phone, syllable or perception based unit (e.g. perceptual centre group) for the task of duration modelling within the speech synthesis framework. Despite the importance of syllable in the temporal and prosodic organisation of speech, we believe that using phone as a building block for our Bayesian model compared to syllable Campbell & Isard (1991) or interperceptual centre group Barbosa & Bailly (1994) has many advantages.

First, there is a data sparsity issue. There are only 40 or so phonemes of English, and there are around 2,000 linguistically allowed syllables that can be generated out of these 40 phones. Creating a database with an adequate coverage of all possible syllables would present a computational challenge.

Second, there is also an issue of redundancy in using syllables instead of phones in duration modelling. Syllable duration is derived from the constituent phone durations, taking into account such factors as type of syllabic nucleus, the number of constituent phones, syllabic stress and position within foot, phrase, utterance, and utterance. However, syllable based duration models only vaguely depend on the constituent phones, thus for example, predicting the same phone durations for the syllables of equal duration. There are linguistic factors that influence timing of speech at various levels.

In addition, phone based duration models such as sums-of-products van Santen (1992*b*) take into account uneven effects of some factors such as stress, within-word, and phrasal position on the duration of the constituent phones. Syllable based models suppose that syllables occurring within the same prosodic and positional context (within-word, within-phrase position) have equal durations (*syllable mediation* hypothesis). Additionally, they assume that syllable duration does not depend on identity of the constituent phones (so called *segmental independence* hypothesis). As it turns out, nei-

ther of the hypotheses were supported by the experimental evidence presented in for example, Shih & van Santen (2000).

1.2.2 Rule based models

Over the past 25 years, there have been a number of models developed for predicting phone duration, ranging from rule-based ones, e.g. by Klatt (1976) to classification and regression tree (CART), described in Breiman, Friedman & Olshen (1984), to sums-of-products (SoP) models by van Santen (1992*b*), van Santen (1994).

In the rule based model by Klatt (1976), the duration of a phonetic segment is modified by successive application of a set of rules. The Klatt model is based on two assumptions. First, all phonetic segments have some inherent duration $D_{inh,P}$ which is independent of phonetic and phonological context. Second, a phone[2] has an absolute minimum duration $D_{min,P}$ "that is required to execute a satisfactory articulatory gesture", as Klatt (1976) puts it. Duration of phone P is therefore described with the following equation:

$$D_j = K \times (D_i - D_{min,P}) + D_{min,P}$$

where P is the name of the phone, K is the parameter of the j-th rule, $D_{min,P}$ is the parameter for the minimum duration assigned to phone P. Initially, D_i is equal to the inherent duration $D_{inh,P}$ of phone P. The values of the parameters for each rule are deduced from a small scale study, and they are manually adjusted during listening tests.

1.2.3 Corpus based models

The progress in computer performance and computer storage devices made it possible to store huge amount of data in computer memory, which was the necessary prerequisite for the development of corpus based duration models. Among these are non-parametric statistical models such as classification

[2]Klatt talks about vowels, though this property equally applies to consonants.

and regression trees (CART) by Breiman et al. (1984) and artificial neural networks (ANN) (e.g. see Bagshaw (1998)), among others. One of the recent duration models that uses statistical analysis combined with the the properties of duration data, is the sums-of-products models (SoP) by van Santen (1992*b*). We will give a full account of the CART and SoP models in Chapters 2 and 3, respectively.

We will talk very briefly about these models here. In the CART model, a phone's duration (absolute or z-score) is predicted by finding the data cluster in the decision tree that matches as many of the feature vector attributes as possible (in the order specified by the tree). In the SoP model, the log of a phone's duration is predicted as a sum of factors' product terms.

1.2.4 Pros and cons of current duration models

Despite being successfully implemented in the MITalk synthesiser described in Allen, Hunnicut & Klatt (1987), rule based models have a few problems. Rule based models do not account for factor interaction. Instead, they rely on manual adjustment of the model's parameters, which eventually makes the analytical representation of the model very complex. The models do not explicitly account for various speaking styles, speaking rates, and dialect differences.

The CART model is easy to build and robust to errors in data. However, CART models can not perform generalisation (interpolation) in cases of rare vectors, hidden or missing data. The performance of CART model degrades when the percentage of hidden (missing) data is too high, as was shown in van Santen (1994).

SoP models use general statistical and mathematical methods such as "ordinal data analysis" by Coombs (1964) and "axiomatic measurement" by Krantz, Luce, Suppes & Tversky (1964) cited in van Santen (1994), that allow the discovery of regular patterns in data. Using these regularities in duration data, SoP models interpolate well in case of rare or hidden (missing)

feature vectors. The models are also robust to noise in the data. One of the problems of SoP models is that the search for a model is a tedious and time consuming process. The number of SoP models is hyper-exponential in the number of variables, i.e. *number of SoP models* $\sim 2^{2^n} - 1$, where n is the number of the model's variables. Therefore, finding a model that fits data the best requires using some heuristic search techniques. Searching for the best SoP model is something of an art and a science.

1.3 Motivation for this thesis

In less than 10 years the focus of the TTS research moved from diphone-based to unit seletion based systems, which do not use explicit duration model, but rather perform the search of the appropriate candidate sequence in a huge database (usually more than 1 hour). One of the main advantages of the approach is the high quality of generated speech. However, this comes at a price: speech databases for unit selection TTS require large storage space and the search through the database is time-consuming. Different methods were used to cut on space and search time ranging from maintaining the cache of the most frequent units (e.g Beutnagel, Mohri & Riley (1999)) to data-driven unit preselecting (Hamza & Donovan (2002)). In addition, unit selection based TTS still have issues with prosodic and spectral discontinuities at the unit joins. To minimise spectral discontinuities Stylianou & Syrdal (2001), Vepa & King (To appear 2005) and others (see for example, Klabbers & Veldhui (1996)) used various join cost metrics. Raux & Black (2003) minimised prosodic discontinuities by selecting portions of F0 contours from a prosodically labelled database. Hofer, Richmond & Clark (2005) added prosodic variety to the synthesised speech using informed blending of prosodically varying databases.

Since the listener's perception of naturalness of synthesised speech is affected by many factors including intonation and duration (see Mayo, Clark & King (2005)) we believe there is still a need for a good duration model.

There are at least two goals that we had in mind when writing this thesis. First, we wanted to explore new approaches to duration modelling in the hope of tackling the problems of the current duration models mentioned in the previous section.

In the book by Jordan (1999), Michael Jordan called graphical models "a marriage between probability theory and graph theory". Bayesian models (a special case of Graphical models), have the same advantages as this more general class of models. Bayesian models allow a model designer to directly embed information about the problem domain variables and their interaction into the model structure. Bayesian models make robust predictions in cases of rare, missing or hidden data. we can combine sets of models into yet another model. Hereby, comes our second motivation for this thesis: we wanted to demonstrate that Bayesian models can be successfully applied to the problem of predicting phone duration, given the many advantages of the approach.

The remainder of this thesis is organised as follows. In Chapter 2 we present the Classification and Regression Trees (CART) model for predicting phone duration. We discuss the basics of the model, followed by a description of the data used in this thesis. Then we describe the CART model training and testing, with the results being discussed. Chapter 3 deals with the sums-of-products model. We discuss the linguistic factors that influence phone duration. Then we describe the details of the SoP model, followed by description of the original SoP model presented by van Santen (1992*b*), van Santen (1994). Then we present our SoP model, discuss its training, testing and results. We present Bayesian models theory in Chapter 4. Starting with the basics of probability theory and Bayesian models, we proceed with describing the junction tree and the junction algorithm for performing Bayesian inference. Following this, we introduce a special type of Bayesian model, namely *Conditional Gaussian* models. We conclude this chapter by discussing learning for Bayesian models. In Chapter 5, we describe two classes of Bayesian models for predicting vowel duration: the *FHLR* models

based on a 4-feature vowel identity representation, and the *FH-compound* model based on a 2-feature vowel identity representation. In the sections of this chapter, we discuss the models' learning, training, and results of predicting vowel duration. In Chapter 6 we define a new class of models that we call *MV-compound* model, whereby consonant identity is represented as manner of production and voice distinctive features. We perform structure and parameter learning on various model topologies. This is followed by Bayesian inference, predicting the durations of the consonants in the test set. We conclude the chapter by discussing the results. Chapter 7 summarises the results presented in this thesis within the Bayesian models framework.

1.4 Publications

Over the course of this work a number of publications appeared. These are Goubanova & Taylor (2000), Goubanova (2001), Goubanova (2003), and Goubanova & King (2005).

Chapter 2

Classification and Regression Tree models

In this chapter we will give an overview of the Classification and Regression Trees (CART) method. We will review the CART model basics in Section 2.1. In Section 2.2 we review various optimisation techniques for building a better tree. We will give the details of the data used in this work in Section 2.3. The baseline CART model for predicting phone duration is described in Section 2.4. The model training is described in Section 2.5. The results of the CART model for predicting phone duration are presented in Section 2.6. We will conclude with a summary of the results in Section 2.7.

2.1 CART model basics

The Classification and Regression Tree method is an example of a machine learning approach that has been used in many areas of business and technology from medical diagnosis to banking to speech technology. CART models are easy to build, robust to errors in data (they perform poorly when the percent of missing data is too high, though). CART models are well suited to problems that require data classification based on the values of attributes (features) that describe separate instances (feature vectors) comprising the

data. For example, our data consists of feature vectors that describe a particular phone depending on its previous and following context, its position within syllable, word, phrase etc. Every feature (attribute) is either discrete, taking on a finite number of values, or continuous, taking on an infinite number of values. Classification trees make predictions about some discrete-valued feature. Regression trees make predictions about some continuous valued feature. For example, in the Festival text-to-speech system described in Black, Taylor & Caley (2000), it is possible to build a classification (decision) tree to predict phrase breaks, or a regression tree to predict phone duration.

CART models are best described in Breiman et al. (1984). We present the basic CART building algorithm implemented in the program called *Wagon*[1], which is a part of the Edinburgh Speech Tools Library described in Black, Caley, King & Taylor (2003). Wagon uses a greedy top-down search through the space of possible decision trees. Wagon implements a *greedy* algorithm in the sense that it chooses the best feature to split the data at each stage of tree building; it never backtracks to re-evaluate earlier choices. Hence, for some data (though not often) the algorithm can build a tree which is suboptimal.

The algorithm starts by choosing the candidate feature for a root of the tree. Each feature is evaluated to determine how well it alone separates the train set. Wagon uses an *impurity* measure to evaluate how similar the feature vectors in the partitions formed by splitting the data using the feature under consideration. The smaller the value of impurity, the less impure the train set is. For regression trees, Wagon uses the variance times the number of feature vectors. For classification trees, Wagon uses the entropy times the number of feature vectors. By multiplying by the number of feature vectors, it is ensured that the procedure does not show favour to smaller partitions. The entropy for some feature (attribute) that takes on c possible values is

[1]We will use the terms *Wagon* and the *CART algorithm* interchangeably, though of course, they are not the same thing.

calculated:

$$H = \sum_{i=1}^{c} -p_i \, log_2 p_i \tag{2.1}$$

where p_i is the proportion of the train set that has the value i for that feature.

The feature with the smallest impurity value (summed across the partitions formed) is chosen as the root of the tree. The data are then split by creating a descendant node for each possible value of that feature. The process is repeated using the feature vectors associated with each subtree to select the best feature to test at that point in the tree. Features that have been incorporated higher in the tree are excluded, so that any feature may appear only once along any path through the tree. The algorithm stops when either every feature has been included along a particular path in the tree, or all the feature vectors associated with a particular leaf have the same target feature value (their entropy is zero). The algorithm also stop when the number of data points in the partition falls bellow some threshold, or the improvement in impurity becomes small from parent to children partitions.

2.2 Building a better tree

Wagon uses different techniques for optimising the tree building process. The goal is not to build a tree that exhaustively classifies every feature vector in the training set. It should rather generalise well on the unseen data. We briefly review some of the methods that allow the building of trees to satisfy this requirement.

In one of the methods the constraints are set on the number of feature vectors in a partition before a question split is considered. Instead of building a full tree with every feature vector classified as a leaf node (this tree of course, will be over-trained), Wagon splits the data if at least some number of feature vectors is left in the partition (this number is called the *stop* value). The stop value of 50 is a usual choice, though smaller or larger numbers may produce better results.

The second method implemented in Wagon is the use of *held-out* data to prune an over-trained tree with a small stop value. Some of the data is taken off to be used for testing a tree. The tree is built using the training set data; it is then pruned back to where it best matches the held-out data. The advantage of this approach is that it allows the stop value to vary through different parts of the tree depending on how general the prediction is when compared against the held-out data.

A more balanced tree can be built, if the stop value is taken to be some percentage (a *balance factor*) of the train set feature vectors. The stop value is then the balance factor times the size of the train set.

2.3 Databases

The data for the present research were derived from 3 Rhetorical databases: 2 RP English voices *rjs* (male) and *lja* (female), and 1 GA English voice *erm* (male). The databases consisted of a set of utterances, one set for each voice. The set of utterances for each voice was divided into train (90%) and test (10%) sets by taking out every 10-th utterance from the set to become a test set, and leaving the rest for the train set. Then the train and test data were dumped as a set of categorical features using Rhetorical's internal tools. The list of features used is shown in Tables 2.1. Further on, the categorical data were transformed into numerical data; for this we used our own data translator written in C++. The data were further separated into vowels and consonants data. The breakdown of the vowel data for the 3 voices is shown in Table 2.2. The breakdown of the consonant data for the 3 voices is shown in Table 2.3.

Name	Example	Name	Example
segment's duration (s)	0.056	*segment in stressed syllable*	*true*
segment's name	*/ax/*	*previous segment in stressed syllable*	*false*
type of a segment	*vowel*	*next segment in stressed syllable*	*false*
syllabic feature	+	*number of segments in syllable*	5
length	*shwa*	*number of segments in next syllable*	2
height	*low*	*segment phrase initial*	*true*
frontness	*back*	*segment phrase medial*	*false*
rounded	+	*segment phrase final*	*false*
manner of production	*fricative*	*previous segment phrase initial*	*true*
place of articulation	*labial*	*previous segment phrase medial*	*false*
voicing	+	*previous segment phrase final*	*false*
previous segment name	*/b/*	*next segment phrase initial*	*false*
previous segment type	*consonant*	*next segment phrase medial*	*true*
previous segment syllabic	−	*next segment phrase final*	*true*
previous segment length	*short*	*segment word initial*	*true*
previous segment height	*low*	*segment word medial*	*true*
previous segment frontness	*mid*	*segment word final*	*false*
previous segment rounded	−	*previous segment word initial*	*false*
previous segment manner	*affricate*	*previous segment word medial*	*false*
previous segment place	*dental*	*previous segment word final*	*false*
previous segment voicing	−	*next segment word initial*	*false*
next segment name	*/l/*	*next segment word medial*	*false*
next segment type	*consonant*	*next segment word final*	*true*
next segment syllabic	+	*segment's syllable position*	*true*
next segment length	*long*	*previous segment's syllable position*	3
next segment height	*mid*	*next segment's syllable position*	2
next segment frontness	−	*onset/nucleus/coda type*	*O1*
next segment rounded	0	*frontness of syllabic vowel*	*back*
next segment manner	*stop*	*number of syllables in word*	3
next segment place	*palatal*	*word class*	*content*
next segment voicing	0	*number of segments in previous syllable*	3

Table 2.1: Linguistic features used to build CART duration model.

2.4 Baseline CART model for predicting phone duration

For our CART duration model we selected a number of linguistic features that describe phones in a particular context. Among the features chosen

Number of vowel tokens			
Voice	**Train**	**Test**	**Total**
lja	35,348	3,876	39,224
rjs	88,997	9,766	98,763
erm	57,104	6,084	63,188

Table 2.2: The number of vowel feature vectors in train, test sets, and the total for the 3 voices: *lja*, *rjs*, and *erm*.

Number of consonant tokens			
Voice	**Train**	**Test**	**Total**
lja	54,489	6,015	60,504
rjs	138,635	14,998	153,633
erm	85,048	9,039	94,087

Table 2.3: The number of consonant feature vectors in train, test sets, and the total for the 3 voices: *lja*, *rjs*, and *erm*.

were the phone's identity, its previous/following segment identity, the stress and accent information, its syllabic, within-word, and within-utterance position, to name just a few. The whole list of features is shown in Table 2.1. There were a total of 64 features selected.

2.5 CART model training

We trained CART models on each of the three voices: *lja*, *rjs*, and *erm*. We trained our baseline CART models with various optional parameters. We experimented with different stop values: 10, 30, 50, 70. In addition, we varied the balance factor: 5%, 10%, 15%, 20%. We also experimented with different amount of held-out data: 5%, 10%, 15%. We trained the models by varying these three optional parameters independently. Having done this, we found the local maxima (minima) of the correlation (RMS error) values on the held-out set for each of the 3 parameters: stop value, balance

factor, and held-out data. We further re-trained the models with these 3 best parameters. The model trained with these parameters was taken to be the best baseline model for each voice.

2.6 CART model results

2.6.1 Vowels

Voice	Stop Value				Held out			Balance factor			
	10	30	50	70	5%	10%	15%	5%	10%	15%	20%
lja	**0.861**	0.856	0.85	0.841	**0.85**	0.848	0.845	0.833	0.845	**0.85**	0.846
rjs	**0.88**	0.874	0.863	0.856	**0.861**	0.861	0.86	0.849	**0.864**	0.86	0.861
erm	**0.890**	0.884	0.876	0.87	0.873	0.870	**0.876**	0.883	**0.885**	0.882	0.88

Table 2.4: The correlation results for the vowels CART model .

Voice	Stop Value				Held out			Balance factor			
	10	30	50	70	5%	10%	15%	5%	10%	15%	20%
lja	**25**	25.3	25.7	26.4	**25.7**	25.9	26.1	27.1	26.9	**25.7**	26.1
rjs	**26**	26.5	27.6	28.3	**27.8**	27.8	27.9	28.9	**27.6**	28	27.8
erm	**26.9**	27.4	27.3	27.1	27.4	27.1	**26.9**	27.1	27.3	**27.0**	27.5

Table 2.5: The RMSE results for the vowels CART model.

We tested our CART models on 2 test sets: one for vowels and one for consonants. In order to do this, we split our original test set into vowels and consonants parts, and did further testing on these separately. The correlation and RMS error results for the CART models tested on the vowels test subset are shown in Tables 2.4-2.5.

It can be seen from Table 2.4, for the *lja* and *rjs* voices the best correlation results were achieved when the models were trained with the stop value of 10, the amount of held-out data of 5%, and the balance factor of 10%-15%. For the *erm* voice, these were a stop value of 10, the amount of held-out data of 15%, and the balance factor of 10%-15% that results in the maximum correlation values. The RMS error results demonstrate similar

behaviour as can be seen from Table 2.5.

2.6.2 Consonants

Voice	Stop Value				Held out			Balance factor			
	10	30	50	70	5%	10%	15%	5%	10%	15%	20%
lja	**0.725**	0.717	0.719	0.710	0.724	**0.727**	0.723	0.721	**0.724**	0.720	0.715
rjs	**0.794**	0.793	0.784	0.780	0.785	**0.789**	0.778	**0.782**	0.779	0.775	0.774
erm	**0.82**	0.813	0.82	0.819	0.820	**0.815**	0.818	0.810	**0.815**	0.813	0.812

Table 2.6: The correlation results for the consonants CART model .

Voice	Stop Value				Held out			Balance factor			
	10	30	50	70	5%	10%	15%	5%	10%	15%	20%
lja	**20.9**	21.1	21.2	21.0	**21.2**	21.3	21.3	21.6	**21.2**	21.4	21.6
rjs	**20.0**	20.4	20.4	20.5	**20.1**	20.4	20.5	20.6	**20.0**	20.2	20.4
erm	**24.0**	24.2	24.1	24.6	24.2	**23.8**	24.2	24.2	**23.9**	24.4	24.5

Table 2.7: The RMS error (ms) results for the consonants CART model.

The correlation results for consonants are shown in Table 2.6. The RMS error results for consonants are shown in Tables 2.7. One can see from Table 2.6, that for the 3 voices, the best correlation results were achieved when the models were trained with a stop value of 10, the amount of held-out data of 10%, and a balance factor of 5%-10%. Since the RMS error results behave similarly, we used the maximum correlation parameters to re-train our baseline models for consonants on the full training set.

2.7 Summary

In this chapter we reviewed the Classification and Regression Trees approach to predicting some discrete or continuous-valued function by classifying the separate instances (feature vectors) based on the values of attributes (features) that describe the data. The CART approach is essentially a search problem; the goal is to find the best tree that describe the data. We reviewed

Phone Type	Correlation			RMS error		
	lja	rjs	erm	lja	rjs	erm
Vowels	0.864	0.84	0.89	25.7	23	26.9
Consonants	0.73	0.794	0.82	21	20	24

Table 2.8: The summary of the CART model correlation and RMS error (ms) results.

the tree searching algorithm implemented in Wagon (described in Black et al. (2003)). Wagon uses various tree search optimisation techniques from limiting the amount of data at the leaf nodes to using different amounts of held-out data for post-pruning to using various stop values at different branches of the tree.

We presented our baseline CART model for predicting phone duration. In order to find the best baseline CART model we experimented with different optimisation parameters. Among the parameters that we experimented with during training were stop value, the amount of held-out data, and the balance factor. As a result of training, we found the best baseline CART models for predicting phone duration for each voice. Table 2.8 shows the best baseline CART models for vowels and consonants for each of the 3 voices. The CART models trained with the stop value of 10, the amount of held-out data of between 5% to 15%, and the balance factor of 10%-15% predict vowel durations in the test set with the maximum correlation (0.84-0.89) and the minimum RMS error (23-26.9 ms). The CART models trained with the the stop value of 10, the amount of held-out data of 10%, and the balance factor of 5%-10% predict consonant durations in the test set with the maximum correlation (0.73-0.82) and minimum RMS error (20-24 ms). These models will be further used for the comparison to the Bayesian models for vowels in Chapter 5 and consonants in Chapter 6.

Chapter 3

Sums of Products (SoP) Models

In this chapter we will introduce the sums-of-products (SoP) model for predicting phone's duration. Most of the information in this chapter is based on the work reported in van Santen (1992*a*), van Santen (1993), and van Santen (1994). We will mention other sources as we go along. In Section 3.1 we discuss the linguistic factors that influence vowel's duration. Next, in Section 3.2 we review the factors that affect a consonant's duration. In Section 3.3 we will introduce the sums-of-products (SoP) model. In Section 3.4 we review van Santen's models for predicting vowel duration. Following this, we define our baseline model for vowels in Section 3.5. In Section 3.6 we review van Santen's models for predicting consonant duration. Next, we define our baseline model for predicting consonant duration in Section 3.7. The two baseline models will be further used for comparison to the Bayesian models for vowels and consonants, described in Chapters 5 and 6 respectively. We conclude this chapter with a summary of results in Section 3.8.

3.1 Linguistic factors influencing a vowel's duration

3.1.1 Literature review

A vowel's duration is influenced by many linguistic factors. Umeda (1975*b*), Klatt (1975) and Crystal & House (1988*a*) reported that vowels in stressed syllables had longer durations than the same vowels in unstressed syllables. Nooteboom (1972), Sluijter & van Heuven (1995), Turk & White (1999), Turk & Shattuck-Hufnagel (2000) reported that the syllables (and consequently a constituent vowel) in accented words were longer than these in de-accented words. van Santen (1992*b*) found the interaction between stress and pitch accent factors: stressed vowels in accented words were significantly longer than non stressed vowels (lengthening percentage of 32% for the male speaker); in de-accented words the difference in duration between stressed and non stressed vowels was smaller but still noticeable (lengthening percentage of 22%). Word initial stressed syllables get shorter as the number of syllables in the word increases as was shown by Lehiste (1972), Klatt (1973), and Port (1981). Moreover, Nooteboom (1972) and Oller (1973) found that stressed vowels in word final syllables had longer durations than vowels in non word final syllables.

In addition, a vowel's duration is affected by its utterance position: the duration of an utterance-final vowel[1] is longer than that of a non utterance final vowel. This effect was reported by Oller (1973), Lehiste (1973), Klatt (1975), Klatt (1976), and Wightman, Shattuck-Hufnagel, Ostendorf & Price (1992).

Peterson & Lehiste (1960), Crystal & House (1988*b*), and van Santen (1992*b*), reported that a vowel's duration depends on voicing and manner of production of the following consonant. van Santen (1992*b*) defined the "standard order" of postvocalic consonant classes arranged in order of as-

[1] Utterance-final vowel is in the utterance-final syllable followed by zero or more consonants

cending vowel duration: *voiceless stops, voiceless affricate, liquids, voiceless fricatives, nasals, voiced stops, voiced affricate*, and *voiced fricatives.*

Given the same linguistic context (e.g. stress and accent status, phrasal position), there is a great variation in durations of different vowels. For example, in the same stress environment the duration of the long vowel such as */oj/* was more than twice longer than the duration of the short vowel such as */I/* as reported by van Santen (1992*b*).

Gregory, Bell, Jurafsky & Raymond (2001) studied the effect of word frequency on duration of content words; they found that the duration of a more frequent word tends to be shorter than the one of a less frequent word. In the similar study of the effect of word frequency on function words Bell, Jurafsky, Fosler-Lussier, Girand, Gregory & Gildea (2003) found that function words tend to be shortened more than content words in a conversational speech.

3.1.2 Factors chosen for vowel analysis

Variable	**# Values**	**Example**
frontness *Front*	3	back
height *Height*	3	mid
length *Length*	4	shwa
roundness *Rnd*	2	rounded
frontness-height *FH*	3	back
within-word position *Wpos*	3	initial
stress *S*	2	stressed
within-utterance position *Utt*	3	utterance medial
following segment identity *Cpos*	10	unvoiced fricative
word class *Wd*	2	function word

Table 3.1: Linguistic variables chosen for predicting vowel's duration.

Based on the results of the research into the variables affecting a vowel's

duration presented in Section 3.1.1, we selected 10 linguistic (causal) variables for predicting a vowel's duration; these are shown in Table 3.1.

Among the variables chosen were the frontness *Front*, height *Height*, length *Length*, roundness *Rnd*, and the compound front-height *FH*, the within-word position *Wpos*, the stress level of a target syllable *S*, the within-utterance position variable *Utt*, the identity of the following segment variable *Cpos*, and the word class *Wd* of the word containing a target vowel.

We represented the vowel identity in two different ways. First, the vowel identity was represented as the set of four variables corresponding to the phonological distinctive features of frontness, height, roundness, and length. The frontness variable *Front* takes on 3 values: *front, medial*, and *back*. The height variable *Height* takes on 3 values: *high, medial*, and *low*. The length variable *Length* has 4 possible values: *short, long, diphthong*, and *shwa*. The roundness variable *Rnd* takes on 2 values: *rounded* and *un-rounded*.

FH values			
	Frontness		
Height	front	medial	back
high	1	2	3
medial	4	5	6
low	7	8	9

Table 3.2: The encoding of the frontness-height compound variable *FH*.

Second, we represented the vowel identity with 2 variables. One is the compound frontness-height variable *FH* based on the phonological distinctive features frontness *Front* and height *Height*. The other is roundness variable *Rnd*. The frontness-height variable *FH* takes on 9 values. The front-height *FH* variable encoding is shown in Table 3.2.

The within-word position *Wpos* variable takes on 3 possible values corresponding to *initial, medial*, and *final* position of the syllable with a target vowel in the word. The stress variable *S* can take 2 values: *stressed* and

unstressed. The within-utterance position variable *Utt* describes phrasal position of the word with a target vowel. It takes on 3 values: *initial, medial,* and *final.*

The identity of the following segment variable *Cpos* takes on 10 values. When the following segment is a consonant, the value of the *Cpos* variable is based on voicing and manner of production features for consonants: *voiceless stops, voiceless affricate, approximants, voiceless fricatives, nasals, voiced stops, voiced affricate, voiced fricatives* and *liquids.* In addition, the

Following segment *Cpos*	**Value**
voiceless stop	1
voiceless affricate	2
liquids	3
voiceless fricatives	4
nasals	5
voiced stops	6
voiced affricate	7
voiced fricatives	8
vowel	9
silence	10

Table 3.3: The encoding of the following segment identity variable *Cpos*

following segment identity variable *Cpos* takes on values *vowel* and *silence.* The values of the *Cpos* variable are shown in Table 3.3.

van Santen (1992*b*) studied the effect of the previous segment identity on the vowel duration; he found that "pre-vocalic consonants pit voiced stops against all other classes". We did not choose the previous segment identity variable for predicting a vowel's duration, as our preliminary experiments had shown this variable had an insignificant effect on a vowel's duration.

To account for the effect of the word frequency on the duration of the word's phones we also considered a word class factor represented by a binary

discrete variable *Wd*, describing whether the word with a target vowel is a content (open class) or a function (closed class). The class of a word, either function or open class, implicitly represents word frequency information.

3.2 Linguistic factors influencing a consonant's duration

3.2.1 Literature review

A consonant's duration is influenced by a number of factors such as the consonant's identity, within-word position, stress level of the previous and following vowels, phrasal position of the word containing the target consonant, its syllabic position, identity of the previous and following segment. van Santen (1994) had found that duration of intervocalic consonants (VCV) depends on the consonant's manner of production and voicing, with voiceless stops being the shortest, and voiced fricatives being the longest in duration. van Son & van Santen (1997) reported on the interaction of manner of production (fricative, plosive, etc.) and voice factors, with voiced consonants durations being longer than those of unvoiced consonants.

A consonant's duration is also affected by within-word position. Oller (1973), Cooper (1991) and Fougeron & Keating (1997) reported that consonant's constriction duration is longer in word-initial than in word-medial position. van Son & van Santen (1997) had found the interaction between within-word position, stress and a consonant identity represented as prime articulator (labial, coronal, post-coronal). Turk & Shattuck-Hufnagel (2000) had found that duration of a primary stressed syllable is affected by the position of a word boundary. In particular, consonants in word-initial primary stressed syllable position were found to be longer, all else being equal.

Stress level of the previous and following vowel affects stop consonant durations such as intervocalic /*t*/ as was shown by Umeda (1977). Pre-stressed consonants are longer than other consonants as reported by Oller

(1973), Klatt (1974), and Umeda (1975*a*). van Santen (1994) and van Son & van Santen (1997) had also found a significant effect of stress and within-word position on intervocalic consonants' durations. Haggard (1973) cited in Klatt (1976) reported consonants being shorter in clusters than in a *CV* environment. van Santen (1994) also found that duration of consonants in clusters is affected by the preceding and following segment identity, syllable boundary (e.g., syllable initial vs. not syllable initial), stress of the previous and following vowel, and accent status of the word.

Consonants in the phrase-final syllable were found to be longer than those in the phrase-medial position as was reported by Oller (1973), Lehiste (1973), Klatt (1975), Klatt (1976), and Wightman et al. (1992), among others.

3.2.2 Factors chosen for consonant analysis

Variable	**# Values**	**Example**
consonant type *C*	24	*/ch/*
manner-voice *MV*	9	voiced fricative
within word position *Wpos*	3	initial
stress *S*	2	stressed
within utterance position *Utt*	3	utterance medial
syllabic position *Syl*	3	coda
previous segment identity *Cpre*	3	consonant
following segment identity *Cpos*	3	silence
frontness of syllabic vowel *Front*	3	front
number of syllables in word *NSyls*	5	3

Table 3.4: Linguistic variables chosen for predicting consonant's duration.

Based on the literature review presented in Section 3.2.1 we selected 10 linguistic variables for predicting consonant's duration. These are shown in Table 3.4. Among the variables chosen were the consonant type *C*, the

manner-voice *MV*, the within-word position *Wpos*, the stress level of the syllabic vowel *S*, the within-utterance position *Utt*, the syllabic position *Syl*, the identity of the previous *Cpre* and following segment *Cpos*, and the frontness of the syllabic vowel *Front*, and the number of syllables in the word containing the consonant *NSyls*.

	Voicing	
Manner	unvoiced	voiced
stop	1	6
affricate	2	7
approximant	3	
fricative	4	8
nasal	5	
tap		6
lateral	9	

Table 3.5: The variable encoding of the manner-voice compound variable *MV*.

We represented consonant identity in 2 different ways. Initially, we encoded consonant identity as a 24-valued (25-valued for *erm* voice) consonant type variable *C*. The encoding for consonant type variable is shown in Table 3.6. Next, we represented the consonant identity with manner of production and voicing distinctive features as a compound manner-voice variable *MV*. The choice of representation followed from the results reported by van Santen (1994) and van Son & van Santen (1997), who found that duration of a consonant depends on its manner of production and voicing. The manner-voice variable *MV* takes on 9 values; and the variable encoding is shown in Table 3.5.

The within-word position variable *Wpos* represents the position of a consonant within the word; it takes on *initial*, *medial*, and *final* values. The stress variable *S* represents the stress level of the syllabic vowel, and takes on *stressed* and *unstressed* values.

GA English	**Encoding**	**RP English**	Encoding
b	1	b	1
ch	2	ch	2
d	3	d	3
dh	4	dh	4
dx	5	f	5
f	6	g	6
g	7	h	7
hh	8	jh	8
jh	9	k	9
k	10	l	10
l	11	m	11
m	12	n	12
n	13	ng	13
ng	14	p	14
p	15	r	15
r	16	s	16
s	17	sh	17
sh	17	t	18
t	18	th	19
th	19	v	20
v	21	w	21
w	22	y	22
y	23	z	23
z	24	zh	24
zh	25		

Table 3.6: Consonant type C variable encoding for RP and GA English voices.

The within-utterance position variable *Utt* describes the phrasal position of the word containing the target consonant; it takes on *initial, medial*, and *final* values. The syllabic position variable *Syl* represents the position of a consonant within a syllable; it takes on the values *onset, coda*, and *syllabic.*

The identity of the previous (following) segment variable(s) *Cpre* (*Cpos*) represents the information about the previous (following) segment in a broad

sense; it takes on 3 values: *consonant, vowel*, and *silence*. The frontness of the syllabic vowel variable *Front* takes on 3 values: *front, medial*, and *back*.

The number of the syllables in a word *NSyls* represents the information about the number of syllables in the word containing the target consonant. It takes on 5 values: *one*, *two*, *three*, *four*,and > 4.

3.3 Sums of products model basics

The sums-of-products (SoP) model is an example of a general linear model whereby segment duration is represented as a sum of variables' product terms that influence segment duration. In the SoP model from (van Santen 1992*b*), segment duration was modelled as a log-transformation of the variables that represent the linguistic factors affecting segment duration.

Let $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ be a set of variables that are chosen for the analysis. We assume all the variables are discrete. Let $\mathbf{x}$ be an instantiation of the set $\mathbf{X}$, such that each variable X_j is assigned a value, i.e. $X_j = x_j;\ j = 1, \cdots, n$. Then the duration of a segment is given:

$$DUR(\mathbf{x}) = \sum_{l \in \mathcal{A}} \prod_{j \in \mathcal{B}_l} K_{lj}(x_j)$$

where set $\mathcal{A}$ is a set of summation terms; $\mathcal{B}_l$ is a set of product terms of the l-th summand; K_{lj}, called "factor scales", correspond to the contribution of the variable $X_j = x_j$ in the l-th summation term to segment duration. We will call a "factor scale" (van Santen (1994), page 103) K_{lj} a *coefficient* for the variable X_j. In general, we treat $K_{lj}(X_j$ as some function defined on the values of variable X_j.

Consider for instance, the rule-based model of segment duration by Klatt (1976). We can demonstrate that it is an example of a sums-of-products model. We briefly discussed this model in the introduction chapter 1. We write duration of a segment P as:

$$D_j = k_{x_j} \times (D_i - D_{min,P}) + D_{min,P} \tag{3.1}$$

where P is the name of the phone, k_{x_j} is a parameter corresponding to the contribution of the the variable X_j having the value x_j to the duration of a segment P, $D_{min,P}$ is the parameter for the minimum duration assigned to phone P. When the rule is applied for the first time, the D_i is equal to the inherent duration $D_{inh,P}$ of phone P. After n-th application of this rule, the duration assigned to segment P in the feature vector $\mathbf{x}$ is written as:

$$DUR(P, \mathbf{x}) = k_{x_1} \times \ldots \times k_{x_n} \times (D_i - D_{min,P}) + D_{min,P} \qquad (3.2)$$

If we change notation slightly, we re-write Equation 3.2 as:

$$DUR(P, \mathbf{x}) = K_{1,1}(x_1) \times K_{1,2}(x_2) \times \ldots \times K_{1,n+1}(x_{n+1}) + K_{2,n+1}(x_{n+1}) \quad (3.3)$$

where $K_{1,n+1}(x_{n+1}) = (D_i - D_{min,P})$, $K_{2,n+1}(x_{n+1}) = D_{min,P}$, and $K_{1,j}(x_j) = k_{x_1}$, where the subscript i in $K_{i,j}$ refers to i-th summation term, and the subscript j refers to the variable X_j. In case of rule based model, $\mathcal{A} = 1, 2$, i.e. there are two summation terms. The sets of product terms are written as: $\mathcal{B}_1 = \{1, 2, \ldots, n+1\}$; $\mathcal{B}_2 = \{n+1\}$.

As attested in van Santen (1994), many of the duration models in the literature can be represented as sums-of-products models: duration models by Lindblom & Rapp (1973), Coker, Umeda & Browman (1973), and Kaiki, Takeda & Sagisaka (1990), to name a few.

3.4 van Santen's model for predicting vowel duration

For his original SoP model van Santen (1992*b*) selected the following linguistic variables: the vowel's identity *V*, the syllabic stress *S*, the accent status of the word *A*, the number of syllables to the word end *Wpos*, the number of consonants preceding the vowel in the word *Wpre*, preceding (following) segment identity *Cpre* (*Cpos*), utterance position *Utt*. Table 3.7 (reproduced from Table 5 in van Santen (1992*b*)) shows the variable names, the number of possible values and example values. As can be seen from the

Variable	# Values	Example
A	3	de-accented
S	3	unstressed
V	9	{I, U}
Wpos	5	word-final closed syllable
Wpre	3	word-initial vowel
Utt	4	final
Cpos	6	unvoiced fricative
Cpre	3	voiced stop

Table 3.7: Linguistic variables selected by van Santen for the SoP model.

table, the vowel identity variable was collapsed to 9 levels, "each containing either one vowel or a group of vowels with similar intrinsic durations" (van Santen (1992*b*), page 539). In this model the lexical stress variable had three levels: *unstressed, primary stressed,* and *secondary stressed.* Accent A had 3 possible values: *accented, de-accented, cliticised.* The following segment identity variable had 6 possible values: *voiceless stops, liquids, voiceless fricatives, nasals, voiced stops, voiced fricatives.* Utterance position variable *Utt* had 4 possible values: *utterance-initial, utterance-medial,* (vowels were at least 5 segments removed from the utterance end), *utterance-penultimate* (word-penultimate vowels in utterance-final words), *utterance-final.*

A vowel's duration is modelled as a log-transformation of linguistic variables that influence a vowel's duration.

$$\begin{aligned} log[DUR(A,S,V,Cpre,Cpos,Wpre,Wpos,Utt)] = \\ K_{1,1}(A) \times K_{1,2}(S) + K_{2,3}(V) + K_{3,4}(Cpre) + K_{4,6}(Wpre) + \\ K_{5,7}(Wpos) + K_{6,5}(Cpos) + K_{7,5}(Cpos) \times K_{7,8}(Utt), \end{aligned} \tag{3.4}$$

where $K_{1,1}(A)$ is a pitch accent A variable coefficient; $K_{1,2}(S)$ is the syllabic stress S variable coefficient; $K_{2,3}(V)$ is the vowel identity V variable coefficient; $K_{3,4}(Cpre)$ is the identity of the preceding consonant $Cpre$ variable coefficient; $K_{6,5}(Cpos)$ and $K_{7,5}(Cpos)$ are the identity of the postvocalic

consonant *Cpos* variable coefficients; $K_{4,6}(Wpre)$ is the preceding number of consonants *Wpre* variable coefficient, $K_{5,7}(Wpos)$ is the number of syllables to the word end *Wpos* variable coefficient; and $K_{7,8}(Utt)$ is the utterance position *Utt* variable coefficient.

3.4.1 Model training and testing

The size of the feature space (i.e. total number of combinations of values for all linguistic variables) according to the model was 87,480. The total number of vowel tokens for the male voice was 18,046. The eight variables of Equation 3.4 accounted for 86% of the variance. Two interaction terms $K_{1,1}(A) \times K_{1,2}(S)$ and $K_{7,5}(Cpos) \times K_{7,8}(Utt)$ were determined using piecewise multiplicative pre-processing followed by residuals analysis (both techniques are described in van Santen (1992*b*)). The model required 32 parameters to be estimated. The correlation and RMS error results are shown

# of tokens of particular type	Correlation	RMS error (ms)
$n > 1$	0.898	31.8
$n > 5$	0.905	30.4
$n > 25$	0.907	27.1

Table 3.8: The correlation and RMS error (ms) results for van Santen's sums-of-products model for vowels ; n is the number of tokens of particular type.

in Table 3.8; these correspond to lines 5-7 taken from Table 6 in van Santen (1992*b*). As can be seen from the table, for feature vectors having at least 2, 6, and 26 observations, the correlation ranges from 0.898 to 0.907. The RMS error ranges from 27.1 to 31.8ms, respectively.

3.5 Baseline model for predicting vowel duration

For our baseline SoP model, we used some of the variables analysed by van Santen (1992*b*). However, for some of the variables we used a different

number of values based on our data. Since our data were not labelled with phrasal accent information, we did not use accent variable A in our model. In addition, the stress variable S took on 2 values, *stressed* and *unstressed*, since in our data there were no distinctions made between primary and secondary stressed vowels. The complete list of the variables that we used for our baseline SoP model for vowels is shown in Table 3.1 on page 21.

Given 2 different vowel identity representations (as a 4-feature and as a 2-feature entity), we defined 2 SoP models for vowels. In one model, the vowel identity was represented with 2 variables: a compound frontness-height variable FH and a roundness variable Rnd. Hence, a vowel's duration is predicted by:

$$\begin{aligned} log[DUR(Wpos, S, Utt, Cpos, FH, Rnd, Wd)] = \\ K_{1,1}(Wpos) + K_{2,2}(S) + K_{3,3}(Utt) + K_{3,4}(Cpos) + \\ K_{4,4}(Cpos) + K_{5,5}(FH) + K_{5,6}(Rnd) + K_{6,7}(Wd) \end{aligned} \tag{3.5}$$

where $K_{1,1}(Wpos)$ is the within-word position $Wpos$ coefficient; $K_{2,2}(S)$ is the stress S coefficient; $K_{3,3}(Utt)$ is the within-utterance position Utt coefficient; $K_{3,4}(Cpos)$ and $K_{4,4}(Cpos)$ are the following segment identity $Cpos$ coefficients; $K_{4,5}(FH)$ is the compound front-height FH coefficient; $K_{4,6}(Rnd)$ is the roundness Rnd coefficient; and $K_{5,7}(Wd)$ is the word class Wd coefficient. We will call this model *SoP-vowels-7*.

In another model, the vowel identity was represented with 4 variables: frontness *Front*, height *Height*, length *Length*, and roundness *Rnd*. Using this model, a vowel's duration is predicted by:

$$\begin{aligned} log[DUR(Wpos, S, Utt, Cpos, Front, Height, Length, Rnd, Wpre, Wd)] = \\ K_{1,1}(Wpos) + K_{2,2}(S) + K_{3,3}(Utt) + K_{3,4}(Cpos) + \\ K_{4,4}(Cpos) + K_{5,5}(Front) + K_{6,6}(Height) + K_{7,7}(Length) + \\ K_{8,8}(Rnd) + K_{9,9}(Wd) \end{aligned} \tag{3.6}$$

where $K_{5,5}(Front)$ is the frontness $Front$ coefficient, $K_{6,6}(Height)$ is the height $Height$ coefficient; $K_{7,7}(Length)$ is the length $Length$ coefficient; and

the rest of the coefficients are defined as in Equation 3.5 We refer to this model as *SoP-vowels-10* model.

3.5.1 Model training and testing

We trained both models *SoP-vowels-7* and *SoP-vowels-10* on the same data described in Section 2.3 on page 12. The models were trained using a standard least-squares method, with the variable encoding specified in Table 3.1 on page 3.1. After we trained each model, we estimated vowel durations in the test set for each of the 3 voices: *lja*, *rjs*, *erm*.

Model Type	**Correlation**			**RMS error**		
	lja	rjs	erm	lja	rjs	erm
SoP-vowels-7	0.71	0.72	0.70	24.5	27.8	32.1
SoP-vowels-10	0.71	0.73	0.70	24.4	27.5	22.2

Table 3.9: The test sample correlation and RMS error (ms) results for our 2 baseline SoP models for vowels by voice.

The test sample correlation and RMS error (ms) results using *SoP-vowels-7* and *SoP-vowels-10* models are shown in Table 3.9. One can see from the table, there are no significant ($p > 0.1, insignificant$) differences in the correlations and RMS errors between the two models: both models *SoP-vowels-7* and *SoP-vowels-10* predict vowel duration with a median (across voices) test sample correlation of 0.71 and a median (across voices) RMS error of 27.5ms. Hence, we chose only one SoP model, namely *SoP-vowels-7* model, for our baseline comparison to the Bayesian models that will be discussed in Chapter 5.

3.6 van Santen's model for predicting consonant duration

3.6.1 Intervocalic consonants

van Santen (1994) performed the analysis of consonants in different segment-level contexts, proposing separate sums-of-products models for intervocalic consonants and consonants in clusters. For intervocalic consonants, van Santen (1994) selected 5 linguistic factors represented as the following variables in the sums-of-products model:

1. Consonant identity C based on voice and manner of production
2. Combined stress S based on the stress values (*primary, secondary, no stress*) of the preceding and following vowels
3. Within-word position *Wpos*
4. Accent status of the word A (*accented, de-accented, cliticised*)
5. Phrasal position *Utt* (*phrase-final*[2] vs. phrase-medial)

He proposed the following model for predicting duration of intervocalic consonants:

$$log[DUR(C, S, Wpos, A, Utt)] = K_{1,[1;3]}(C, Wpos) + K_{2,[1;3]}(C, Wpos) \times K_{2,2}(S) + K_{3,4}(A) + K_{4,5}(Utt) \quad (3.7)$$

where $K_{1,[1;3]}(C, Wpos)$ and $K_{2,[1;3]}(C, Wpos)$ are the coefficients for the compound "consonant identity - word position" variable, the square brackets $[.;.]$ stand for the set of variables selected for analysis, $K_{2,2}(S)$ is the stress S coefficient, $K_{3,4}(A)$ is the accent A coefficient and $K_{4,5}(Utt)$ is the phrasal position variable *Utt* coefficient.

[2]Phrase-final is defined as separated by one vowel and zero or more consonants from the phrase boundary

3.6.2 Consonants in clusters

For consonants in clusters, the models were further distinguished by syllabic and phrasal position variables. In addition, due to data sparsity problems stress and accent variables were collapsed into a compound *stress-accent S-A* variable. For consonants in *syllable onsets* the variables analysed were:

1. Following segment identity *Cpos*
2. Combined preceding segment *Cpre* and syllable boundary *SylBnd* variable (*word-initial, syllable-initial, not syllable initial*)
3. Combined stress-accent of the last vowel *S-A-last* and syllable boundary *SylBnd*
4. Stress-accent *S-A-next* of the next vowel variable

Duration of consonant onsets is written:

$$\begin{aligned} log[DUR(Cpos, SylBnd, Cpre, S-A, S-A-next, S-A-last)] = \\ K_{1,1}(Cpos) + K_{2,[2;3]}(Cpre, SylBnd) + \\ K_{3,[2;6]}(SylBnd, S-A-last) + K_{4,5}(S-A-next) \end{aligned} \tag{3.8}$$

For consonants in *phrase-medial codas* the variables analysed were:

1. Following segment identity *Cpos* × *syllable boundary* (all combinations of segment classes with *word-final* vs.*syllable-final* vs. *not syllable final*)
2. Preceding segment identity *Cpre*
3. Stress-accent of the last vowel *S-A-last*
4. Stress-accent *S-A* of the next vowel × *syllable boundary* (all combinations of *stress-accent S-A* with *word-final* vs.*syllable-final* vs. *not syllable final*)

Duration of phrase-medial codas duration is written as:

$$\begin{aligned} log[DUR(Cpos, SylBnd, Cpre, S-A, S-A-next)] = \\ K_{1,[1;2]}(Cpos, SylBnd) + K_{2,3}(Cpre) + \\ K_{3,4}(S-A) + K_{4,[2;5]}(SylBnd, S-A-next) \end{aligned} \tag{3.9}$$

For consonants in *phrase-final codas* the variables analysed were:

1. Following segment identity *Cpos* × *syllable boundary* (all combinations of segment classes with *word-final* vs.*syllable-final* vs. *not syllable final* vs. *silence*)

2. Preceding segment identity *Cpre*

3. Stress-accent of the last vowel *S-A*

Due to data sparsity problems van Santen (1994) used simple multiplicative models for consonants in all phrasal contexts. Duration of phrase-final codas is written as:

$$\begin{aligned} log[DUR(Cpos, SylBnd, Cpre, S-A-last)] = \\ K_{1,[1;2]}(Cpos, SylBnd) + \\ K_{2,3}(Cpre) + K_{3,3}(S-A-last) \end{aligned} \tag{3.10}$$

3.6.3 Results

We summarise the results of van Santen's SoP model for consonants in Table 3.10 (see van Santen (1994) for details). As can be seen from the table, all the models predict consonant's duration with correlation above 0.8 even when the number of consonant tokens in the training set is below 2,000. We compare van Santen's results to our baseline SoP model results in Section 3.7 and further in Section 3.8.

Model Type	Correlation	# Tokens
Intervocalic (VCV) Consonants (overall)	0.903	10,420
Phrase-medial VCV	0.907	8,725
Phrase-final VCV	0.887	1,695
Onsets	0.841	7,523
Phrase-medial codas	0.824	8,188
Phrase-final codas	0.871	2,025

Table 3.10: The results for van Santen's model for predicting consonant's duration. The correlation results by model type and phrasal contexts.

3.7 Baseline model for consonants

Initially, we defined 2 SoP models for consonants. For the first model, we selected 5 linguistic factors: the consonant type C, the within-word position *Wpos*, stress level of the syllabic vowel S, the number of syllables in the word *NSyls*, and the frontness of the syllabic vowel *Front*. We refer to this model as *SoP-cons-5*.

For the second model, we chose the following 8 variables: manner-voice MV, the within-word position *Wpos*, the stress level of the syllabic vowel S, the within-utterance position *Utt*, the syllabic position variable *Syl*, the identity of the previous *Cpre* and following *Cpos* segments, and frontness *Front* of the syllabic vowel. We refer to this model as *SoP-cons-8*.

The variable names, the number of possible values, and example values for both models are shown in Table 3.4. For simplicity we decided to use a simple multiplicative model[3]. Using the *SoP-cons-5* model, consonant duration can be predicted by:

$$log[DUR(C, S, Wpos, NSyls, Front)] = \tag{3.11}$$

$$K_{1,1}(C) + K_{2,2}(S) + K_{3,3}(Wpos) + K_{4,4}(NSyls) + K_{5,5}(Front) \tag{3.12}$$

[3]A multiplicative model becomes an additive model in log-domain.

Using the *SoP-cons-8* model, consonant duration can be predicted by:

$$log[DUR(MV, S, Wpos, Utt, Syl, Cpre, Cpos, Front)] = \tag{3.13}$$

$$K_{1,1}(MV) + K_{2,2}(S) + K_{3,3}(Wpos) + K_{4,4}(Utt) + K_{5,5}(Syl) + K_{6,6}(Cpre) + K_{7,7}(Cpos) + K_{8,8}(Front) \tag{3.14}$$

3.7.1 Model training and testing

Similar to the SoP models for vowels, we trained the models for consonants using a standard least-squares method, with the variable encoding specified in Table 3.4. We trained model *SoP-cons-5* on part of the *rjs* voice, since it was a pilot study that will be further described in Chapter 6. We trained the model *SoP-cons-8* on each of the 3 voices: *lja*, *rjs*, *erm*. After we trained each model, we estimated the durations of the consonant tokens of the test set.

Model Type	**Correlation**			**RMS error**		
	lja	rjs	erm	lja	rjs	erm
SoP-cons-5	0.73			27.3		
SoP-cons-8	0.74	0.79	0.76	25	26	33

Table 3.11: The test sample correlation and RMS error (ms) results for 2 candidate SoP models for consonants, by voice.

Table 3.11 shows the correlation and RMS error (ms) results for the two models. As can be seen from the table, the *SoP-cons-8* model gives better predictions than the *SoP-cons-5* model both in terms of the correlation and RMS error, with a median (across voices) correlation of 0.76 for *SoP-cons-8* against 0.73 for *SoP-cons-5*. In terms of the RMS error, the *SoP-cons-8* model also predicts consonant duration with a lower median RMS error of 26ms (against 27.3ms for the *SoP-cons-5* model). Given these overall results, we decided to use the model *SoP-cons-8* as our sums-of-products baseline model for consonants.

3.8 Summary

Model Type	Correlation	RMSE (ms)
van Santen's vowels	0.898-0.907	27.1-32.8
our SoP-vowels-7	0.70-0.72	24.5-32.1
van Santen's consonants	0.824-0.907	*NA*
our SoP consonants	0.74-0.79	25-33

Table 3.12: The summary of the sums-of-products correlation and RMS error (ms) results. The RMS error results for van Santen SoP model for consonants is marked *NA* (not available).

In this chapter, we discussed the sums-of-products model for predicting phone duration. In particular, we described SoP models for two broad classes of phones: vowels and consonants. The summary of the results for both classes of models is presented in Table 3.12.

For vowels, we reviewed the original model by van Santen (1992*b*). First, we discussed the linguistic factors he chose for the analysis (they are listed in Table 3.7). Given these factors, vowel's duration is predicted using Equation 3.4. The summary of van Santen's results for predicting vowel duration is shown in Table 3.8.

We then discussed the linguistic variables that we selected for our baseline SoP model for vowels. In particular, we introduced the word class variable that was not present in the original model for vowels. In addition, we did away without the accent variable, since our data were not labelled with word accent status information. We also considered two different vowel identity representations. One was a 4-feature representation based on frontness, height, length, and roundness distinctive features. The other was a 2-feature representation based on compound front-height and roundness variables. Given these changes in the set of variables under consideration, we defined two baseline models: *SoP-vowels-7* and *SoP-vowels-10*. Using these models, vowel's duration is predicted as in Equations 3.5 and 3.6.

Since the results for both models did not differ much in terms of the correlation and RMS error values as is evident from Table 3.9, we chose the model *SoP-vowels-7* as our baseline model for vowels.

We compared the performance of our baseline *SoP-vowels-7* model to van Santen's model for vowels. In terms of the correlation, the van Santen model performs better (0.898-0.907) than our baseline *SoP-vowels-7* model (0.70-0.72), as can be seen from the summary table. In terms of the RMS error however, our baseline *SoP-vowels-7* model (24.5-32.1ms) performs better than van Santen's model (27.1-32.8) for vowels.

We also discussed van Santen's models for consonants. He suggested to use separate models for intervocalic consonants and consonants in clusters, with different set of linguistic variables considered for each model class. Furthermore, for consonants in clusters he made even finer distinctions based on the syllabic and phrasal position of the target consonant. Consequently, there were 4 different models defined: intervocalic consonants, consonants in syllable onsets, phrase-medial codas, and phrase-final codas models. Table 3.12 shows just the range of the correlation and RMS error values from the worst (e.g. 0.824 for the phrase-medial codas model) to the best (0.907 for the intervocalic consonants model).

Following the discussion of the van Santen's model for consonants, we described our baseline model for consonants. In fact, there were two models considered: the model *SoP-vowels-5* based on 5 linguistic variables, and the model *SoP-vowels-8* based on 8 linguistic variables. For both models we considered the consonant identity, the within-word position, the stress and frontness of the syllabic vowel variables. For the *SoP-vowels-5* model we also considered the number of syllables in the word variable. In addition, in this model the consonant identity was represented as a 24-valued consonant type variable.

For the *SoP-vowels-8* model, we also selected manner-voice, within-utterance and syllabic position variables and the identity of the previous and following segments variables. In this model, the consonant identity was

represented as the manner-voice variable. As it turned out the *SoP-cons-8* model predicted consonant duration with higher correlation values (0.79 vs. 0.73), and lower RMS errors (26ms vs. 27.3ms). For that reason, we selected the *SoP-cons-8* model as our baseline model for consonants.

We further compared van Santen's model for consonants to our baseline *SoP-cons-8* model. As one can see from the summary table, the *SoP-cons-8* model compares well against van Santen's models for consonants. It predicts consonant duration with a test sample correlation that is lower than those for van Santen's models (0.79 vs. 0.907 for the best results).

It should be pointed out however, that direct comparison of the van Santen's and our baseline models is difficult for various reasons. First, we used different set of variables for our baseline models. For example, instead of using a vowel identity variable based on inherent duration as did van Santen, we used a bundle of 4 distinctive features: frontness, height, length, and roundness to define vowel identify variables. Second, we used different data recorded from different speakers and accents. Third, we did not compare the performance of all possible models, since it would be computationally infeasible for the task; we rather chose the best linguistically motivated model. We contend that our baseline SoP models for vowels and consonants are no worse than the original models defined in van Santen (1992*b*) and van Santen (1994). And hence, they can be used for adequate comparisons to the corresponding Bayesian models that will be discussed in Chapters 5 and 6.

Chapter 4

Bayesian Belief Networks

In this chapter we will give a brief overview of Bayesian Belief Networks (throughout the thesis we will refer to Bayesian Belief Networks as Bayesian networks, Bayesian models or just BNs), highlighting the points that are of particular interest for the problem of segment duration prediction in text-to-speech synthesis.

The information presented below is compiled from various sources; as a basis for this presentation we mainly used Cowell, Dawid, Lauritzen & Spiegelhalter (1999), Olesen (1993), Jordan (1999), Huang & Darwiche (1996), and Lauritzen & Spiegelhalter (1988). We will mention other sources as we go along. In Section 4.1 we will briefly remind the reader the basics of probability theory and introduce Bayesian networks. In Section 4.2 we will give an overview of a junction tree, a secondary structure built from a Bayesian Network. We will review the procedures for building a junction tree from a Bayesian network in Section 4.3. We will talk about a special case of Bayesian Networks, namely Conditional Gaussian networks (that will be used for segment duration prediction), in Section 4.4. We will proceed with describing inference algorithms in Section 4.5. and structure learning algorithms for BNs in Section 4.6. We conclude this chapter by discussing the BN parameter learning procedure in Section 4.7.

4.1 Probability and Bayesian Networks basics

4.1.1 Probability

Before we can proceed with discussing Bayesian networks, we need to talk about notational conventions. Let us denote random variables with upper case letters, for example X, Y, Z. Sets of variables are denoted with bold face letters, for example $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. A variable X is *instantiated* if it is assigned a value x from a finite set (for discrete variables) or an infinite set of real numbers (for continuous variables). The instantiation of every variable in a set is denoted with a bold face lower case letter, e.g. $\mathbf{x}$.

The theory of Bayesian Networks relies heavily on the concepts of uncertainty and probability as a numerical measure of the degree of belief about some random event given the data at hand. Consequently, the axioms of probability and Bayes' theorem form the basis of Bayesian analysis. For completeness, we will present the axioms of probability and Bayes' theorem below.

The **probability** of an event A denoted $P(A)$, is a number in the interval [0, 1] that describes our degree of belief about an event A. Two important concepts of probability theory are those of *conditional* and *marginal* probabilities. *Conditional* probability is essentially the statement of the form: given the event $B = b$ the probability of the event $A = a$ is equal to $\alpha; 0 < \alpha < 1$ written as $P(A = a|B = b) = \alpha$. The event $B = b$ is called *evidence* or an *observation.* The probability of the event $A = a$ with no evidence is called the *marginal* probability or just probability; it is written as $P(A = a)$. The axioms of probability state that

1. $P(A = a) = 1$ if and only if the event A=a is observed

2. If event A occurs when event B does not occur (mutually exclusive events), then

$$P(A \text{ or } B) = P(A) + P(B)$$

3. The joint probability of two events A and B occurring is given by the product rule.

$$P(A, B) = P(A|B)P(B)$$

We can express the joint probability distribution of two events A and B via marginal and conditional probabilities:

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A)$$

From there, Bayes' formula follows:

$$P(A|B) = P(A) \times \frac{P(B|A)}{P(B)} \tag{4.1}$$

The interpretation of Bayes' formula is pretty straightforward. We start with our prior belief about the event A represented by a *prior* probability $P(A)$. Then we observe event B. Bayes' theorem 4.1 says that our revised belief about A represented by the *posterior* probability $P(A|B)$, is obtained by multiplying the prior $P(A)$ by the ratio $\frac{P(B|A)}{P(B)}$. The quantity $P(B|A)$, which is a function of A for a fixed B, is called the *likelihood* function. Schematically Bayes' formula 4.1 can be written like this.

$$\textbf{posterior} \propto \textbf{prior} \times \textbf{likelihood}$$

4.1.2 Bayesian Networks defined

Before we proceed with the Bayesian networks definition, we will need some definitions from the graph theory. A graph is a pair $G = (\mathbf{U}, \mathbf{E})$, where $\mathbf{U} = \{X_j | j = 1, \ldots, n\}$ is a finite set of vertices, and $\mathbf{E} = \{(X_j, X_k) \mid j, k = 1, \ldots, n\}$ is a subset of the set $\mathbf{U} \times \mathbf{U}$ of ordered pairs of vertices. If each vertex in a pair is distinct, the graph is called *simple*, i.e., there are no multiple edges or loops. A graph G is called *undirected* if, for every pair of vertices X_j, X_k the $(X_j, X_k) \in \mathbf{E}$, then the edge also belong to the set E: $(X_k, X_j) \in \mathbf{E}$. Otherwise, a graph is called *directed.* A graph G is called *directed acyclic graph* (or DAG), if it is directed, simple and does not contain direct cycles.

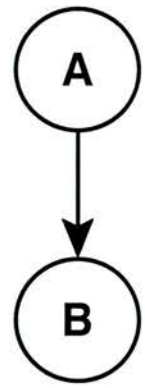

Figure 4.1: Bayesian network example

A **Bayesian network** is specified by a pair (G, P), where graph G represents a qualitative (graphical) part and P represents a quantitative (probabilistic) part. The graphical part of a network G, encodes the information about the problem domain variables and relations between them. Suppose we have a finite set of random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, which is also called the *problem domain set* or the *universe* of variables. The variables of $\mathbf{X}$ are represented as vertices in a directed acyclic graph (DAG). The dependency relations among the variables of $\mathbf{X}$ are represented as edges in a DAG. Throughout the thesis we will be using the terms *variable, vertex*, and *node* interchangeably. Likewise, we will be using the terms *edge* and *link* interchangeably. A simple BN is shown in Figure 4.1. In a directed graph an edge is graphically represented with an arrow going from one vertex to the other, for example, an edge $A \rightarrow B$ as shown in Figure 4.1. Given a directed edge $A \rightarrow B$, vertex A is called a *parent*, and vertex B is called a *child.* A set of parents of a node B is denoted $\mathbf{Pa}(B)$. The instantiation of the parents of node B is denoted $\mathbf{pa}(B)$. A node B and its parents $\mathbf{Pa}(B)$ are called the *family* of B, i.e. $F_B = B \cup \mathbf{Pa}(B)$.

The joint probability distribution (JPD) over the variables of a BN $P(\mathbf{X})$ quantifies the dependency relations among the variables of a network. In addition, for each node there exists a conditional probability distribution (CPD), $P(X_j|\ \mathbf{Pa}(X_j))$; conditioning being on the parents of the node X_j.

As it turns out, Bayesian networks allow for a more compact representation of $P(\mathbf{X})$ by using the *directed Markov property*; this says that a variable is conditionally independent of its non-descendants $nd(X)$ given its parents. This property can be formally written like so:

$$X \amalg nd(X)|\mathbf{Pa}(X) \tag{4.2}$$

where $nd(X)$ are non-descendent's of the variable X. This allows us to factorise the joint probability distribution $P(\mathbf{X})$ into a set of local conditional distributions over variables of a network. Given a set of problem domain variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ and a set of conditional probability distributions associated with every node in a BN, the joint probability distribution $P(\mathbf{X})$ factorises thus:

$$P(\mathbf{X}) = P(X_1, X_2, ..., X_n) = \prod_{j=1}^{n} P(X_j|\mathbf{Pa}(X_j)) \tag{4.3}$$

where n is the size of the BN, $\mathbf{Pa}(X_j)$ is the set of parents of node X_j. The goal of Bayesian inference is to calculate the marginal distributions given one or more observed variables. However, the direct use of the model's JPD $P(\mathbf{X})$ to calculate these marginals is intractable, since the size of the JPD $P(\mathbf{X})$ is exponential in the number of the variables in the network. For example, the size of $P(\mathbf{X})$ of a BN consisting of n binary nodes is 2^n. Hence, it may become huge for larger networks (size 20 or larger). To demonstrate the idea of inefficiency of the direct JPD calculation let us consider a simple example borrowed from Jordan (1999) (see Figure 4.2).

Suppose we are given a network of 3 nodes X, Y, Z; we know the probabilities being $P(X), P(Y, X)$ and $P(Z, X)$. We observe $Y = y$. The task is to calculate $P(Z|Y = y)$. The straightforward method is performed like this.

1. Calculate the joint probability $P(X, Y, Z) = P(Y = y\,|X, Z)P(Z|\,X)P(X)$
2. Calculate the marginal $P(Y = y) = \sum_{X,Z} P(X, Y = y, Z)$
3. Calculate the marginal $P(Z, Y = y) = \sum_{X} P(X, Y = y, Z)$

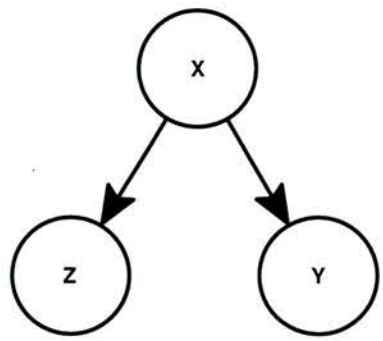

Figure 4.2: Bayesian network example

4. Calculate $P(Z|Y = y) = P(Z, Y = y)/P(Y = y)$

By exploiting the local structure of the network in question, the calculations can be made more efficient.

1. By using Bayes' formula 4.1 calculate $P(X|\ Y = y) = \frac{P(Y=y|\ X)P(X)}{P(Y=y)}$ where $P(Y = y) = \sum_X P(Y = y|X)P(X)$

2. Finally, calculate $P(Z|Y = y) = \sum_X P(Z|\ X)P(X|\ Y = y)$

If we assume each of the variables can take on 10 values, the former method of the JPD $P(X, Y, Z)$ calculation requires a CPD table of 1000 parameters, whereas the latter requires a CPT of at most 100 parameters. For larger networks this can substantially reduce the number of parameters being calculated, and consequently, save the computation time and storage space[1].

[1]For example, the JPD for a network of size 30 with all the nodes being binary consists of 1,073,741,824 entries.

4.2 Junction tree basics

4.2.1 Potential function defined

As was demonstrated in Section 4.1, a simpler representation of a Bayesian network (a joint probability distribution $P(\mathbf{X})$ and a graphical structure G) is needed in order to make the computations more efficient. This is done by transforming a Bayesian network into a *junction tree.* The graphical structure G of a network is transformed into a set of clusters (cliques) of the nodes $\mathbf{X}$. (We define a *clique* to be a subset of nodes which are fully connected and maximal, i.e. no additional node can be added to the subset so that it remains fully connected.) The joint probability $P(\mathbf{X})$ of a network, represented as the product of the local CPDs attached to each node in the graph G, is expressed via the product of *potential* functions defined over the cliques of a junction tree.

A **potential** $\phi_{\mathbf{X}}$ is a function defined on the set of variables $\mathbf{X}$ such that for every instantiation of a set variables $\mathbf{x}$ there exists a non-negative real number $\alpha_{\mathbf{x}} \in \mathcal{R}$.

$$\alpha_{\mathbf{x}} = \phi_{\mathbf{X}}(\mathbf{x})$$

The number $\alpha_{\mathbf{x}}$ that $\phi_{\mathbf{X}}$ maps $\mathbf{x}$ into is called an *element.*

In general, there are operations of *marginalisation* and *multiplication* defined on potentials[2]. Suppose we have two sets of variables $\mathbf{X}$ and $\mathbf{Y}$ such that $\mathbf{Y} \subseteq \mathbf{X}$, with the potential $\phi_{\mathbf{X}}$ being defined on $\mathbf{X}$. We define the **marginalisation** of $\phi_{\mathbf{X}}$ into $\mathbf{Y}$ to be a potential $\phi_{\mathbf{Y}}$ such that for each element $\phi_{\mathbf{Y}}(\mathbf{y})$:

$$\phi_{\mathbf{Y}}(\mathbf{y}) = \sum_{\mathbf{X}\setminus\mathbf{Y}} \phi_{\mathbf{X}} = \phi_{\mathbf{X}}(\mathbf{x_1}) + \phi_{\mathbf{X}}(\mathbf{x_2}) + \cdots$$

where $\mathbf{x_1}, \mathbf{x_2} \cdots$ are the instantiations consistent with $\mathbf{y}$. The marginal potential ϕ is denoted with a symbol $\sum_{X\setminus Y} \phi$, with Y being the set of

[2]There exist other operations defined on potentials that will be discussed in Section 4.4.2

variables marginalised to, and $X \backslash Y$ being the set of variables marginalised over.

Given two sets of variables $\mathbf{X}$ and $\mathbf{Y}$ with the potentials $\phi_{\mathbf{X}}$ and $\phi_{\mathbf{Y}}$, we define the **multiplication** of $\phi_{\mathbf{X}}$ and $\phi_{\mathbf{Y}}$ to be a potential $\phi_{\mathbf{Z}}$, where $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$, such that for each element $\phi_{\mathbf{Z}}(\mathbf{z})$ consistent with $\mathbf{x}$ and $\mathbf{y}$ it holds:

$$\phi_{\mathbf{Z}}(\mathbf{z}) = \phi_{\mathbf{X}}(\mathbf{x})\phi_{\mathbf{Y}}(\mathbf{y})$$

The multiplication of potentials is denoted as $\phi_{\mathbf{Z}} = \phi_{\mathbf{X}}\, \phi_{\mathbf{Y}}$.

4.2.2 Junction tree defined

Given a Bayesian belief network over a set of variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, a **junction tree** is an undirected tree $\mathcal{T} = (\mathbf{C}, \mathbf{S}, \Phi)$, where $\mathbf{C}$ is a set of *cliques* $\mathbf{C} = \{\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_m}\}; m < n$ such that each clique $\mathbf{C_k}$; $k = 1, \cdots, m$ of the tree $\mathcal{T}$ is a subset of the original problem domain $\mathbf{X}$; $\mathbf{S} = \{S_1, S_2, \cdots, S_l\}$; $l = 1, \cdots, m-1$ is a set of junction tree edges, called *separators*, labelled with the intersection of adjacent cliques; and $\Phi = (\phi_{\mathbf{C}} \cup \phi_{\mathbf{S}})$ is a set of belief potentials defined over the cliques and separators of the junction tree (Φ is also called a **charge** on $\mathcal{T}$). The cliques in $\mathcal{T}$ satisfy *the join tree property*; given two cliques $\mathbf{C_{k_1}}$ and $\mathbf{C_{k_2}}$; $k_1 \neq k_2$ in $\mathcal{T}$, all cliques on the path between $\mathbf{C_{k_1}}$ and $\mathbf{C_{k_2}}$ contain $\mathbf{C_{k_1}} \cap \mathbf{C_{k_2}}$. In addition, for each variable $X_j \in \mathbf{X}$ $(j = 1, \ldots, n)$ the family F_{X_j} is included in at least one of the cliques.

It follows from the direct Markov property of a BN (equation 4.2) that the JPD $P(\mathbf{X})$ can be expressed as the product of potentials, with each potential being a function of a BN variable X_j and its parents $\mathbf{Pa}(X_j)$ like so.

$$P(\mathbf{X}) = \prod_{X_j \in \mathbf{x}} P(X_j \,|\mathbf{Pa}(x_j)) = \phi(x_j, \mathbf{Pa}(X_j)) \tag{4.4}$$

where $\phi(X_j, \mathbf{Pa}(X_j)) = P(X_j|\mathbf{Pa}(X_j))$.

In order to perform local computations within cliques and between neighbouring cliques, the potentials should satisfy the following constraints.

- The junction tree should be *locally consistent*, that is for each clique $\mathbf{C}$ and a neighbouring separator $\mathbf{S}$, it holds:

$$\sum_{\mathbf{C}\setminus\mathbf{S}} \phi_{\mathbf{C}} = \phi_{\mathbf{S}} \tag{4.5}$$

- The joint probability $P(\mathbf{X})$ is encoded via the clique and separator potentials according to the equation:

$$P(\mathbf{X}) = \frac{\prod_k \phi_{\mathbf{C_k}}}{\prod_l \phi_{\mathbf{S_l}}} \tag{4.6}$$

where $k = 1, \cdots, m$, $l = 1, \cdots, m-1$, m is the number of cliques; and $\phi_{\mathbf{C_k}}$ and $\phi_{\mathbf{S_l}}$ are clique and separator potentials, respectively.

From the definition of a junction tree it follows that for each clique $\mathbf{C}$ (or separator $\mathbf{S}$), it holds $\phi_{\mathbf{C}} = P(\mathbf{C})$ or $\phi_{\mathbf{S}} = P(\mathbf{S})$. Consequently, for any variable X_j contained within the clique $\mathbf{C}$ or the separator $\mathbf{S}$ the marginal distribution is given:

$$\begin{aligned} P(X_j) &= \textstyle\sum_{\mathbf{C}\setminus\{X_j\}} \phi_{\mathbf{C}} \\ &= \textstyle\sum_{\mathbf{S}\setminus\{X_j\}} \phi_{\mathbf{S}} \end{aligned} \tag{4.7}$$

4.3 Constructing a junction tree from a Bayesian network

Now we discuss the steps to construct a junction tree from a Bayesian network. Suppose we have the network of 6 nodes shown in Figure 4.3. The operation of transformation of a BN into a junction tree is performed in stages.

1. Moralisation
2. Triangulation
3. Building the junction tree

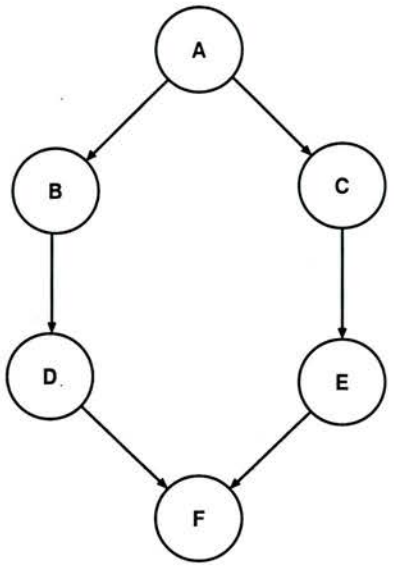

Figure 4.3: Bayesian network with the variable set $\mathbf{X} = \{A, B, C, D, E, F\}$

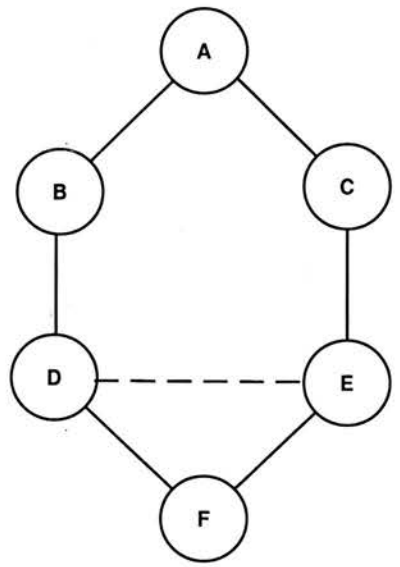

Figure 4.4: The graph with the node set $\mathbf{X}$ transformed into a moral graph G^m.

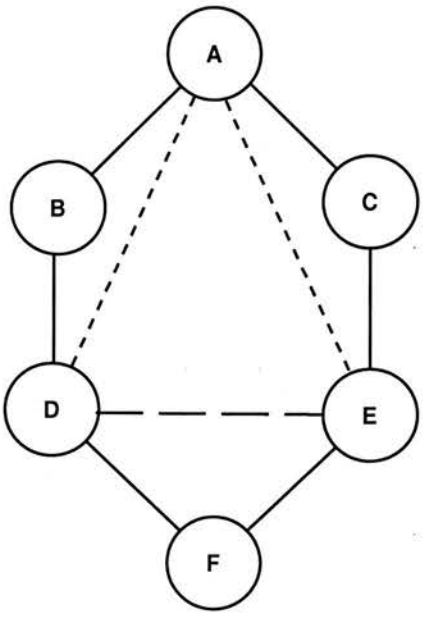

Figure 4.5: Triangulated graph resulting from the moral graph G^m.

First, the graph G of the network is transformed into a *moral* graph G^m by adding an edge between every pair of a node's unconnected parents, e.g. an edge $D \rightarrow E$ is added to the original graph G as shown in Figure 4.4. The directionality of the edges of the graph is dropped (the graph becomes undirected). After the moralisation step, we have an undirected graph in which each node and its parents form a complete subgraph of a graph G^m.

Next, from the moral graph G^m we create a triangulated graph G^t by adding sufficient additional edges between the nodes so that there are no cycles of length 4 or more distinct nodes without a short-cut. In Figure 4.5 the graph G^t resulting from the triangulation of the moral graph G^m of Figure 4.4 is shown.

In general, the problem of finding an optimal triangulation is $\mathcal{NP}$-hard. However, there have been a number of heuristic algorithms developed that are examined in detail in Kjaerulff (1992). The basic algorithm runs as follows: given an ordering of the nodes $\pi = (X_1, X_2, \cdots, X_n)$ of the graph G^m, we can recursively choose the node X_j in reverse order, beginning with X_n, and join it with all its neighbours that appear earlier in the ordering π

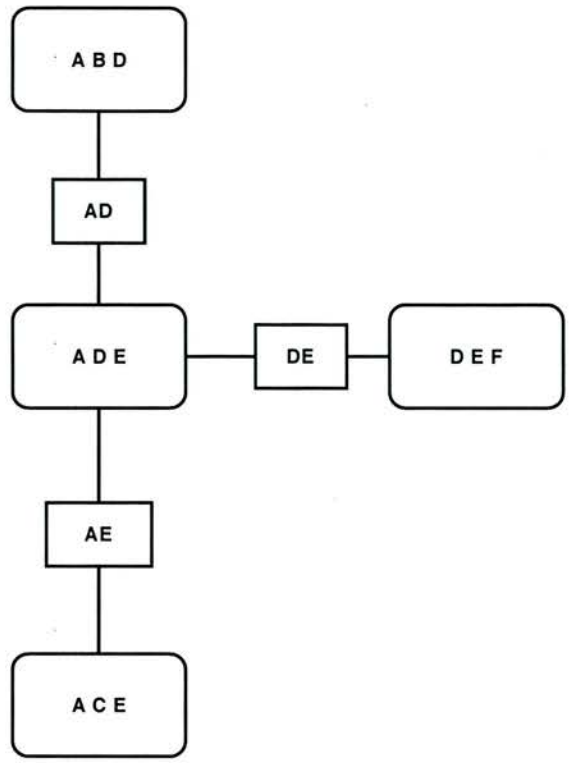

Figure 4.6: Junction tree built from the network of Figure 4.3

and are not yet joined (the node X_j and its neighbours will form an *induced cluster*). Therefore, the problem of finding a good triangulation is equivalent to the problem of finding a perfect ordering[3].

Finally, once the cliques of a triangulated graph are identified, a junction tree can be built. The cliques of the triangulated graph G^t can be identified during the process of triangulation; each induced cluster that is not a subset of any previously saved cluster forms a clique. Thus identified cliques are joined together by applying the running intersection property to form a junction tree. An example junction tree built from the graph of Figure 4.3 is shown in Figure 4.3. The set of cliques of the junction tree consists of four cliques ABD, ADE, ACE, and DEF (each clique is labelled with the nodes it contains), i.e. $\mathbf{C} = \{ABD\} \cup \{ADE\} \cup \{ACE\} \cup \{DEF\}$. The

[3]We call an ordering of a graph *perfect* if the parents of every node form a complete set.

separators are AD, AE, and DE, i.e. $\mathbf{S} = \{AD\} \cup \{AE\} \cup \{DE\}$.

4.4 Bayesian Networks for predicting segment duration

How does our goal of segment duration prediction fit into the Bayesian framework? On the one hand, Bayesian networks provide a model representation for the joint distribution of a set of variables in terms of conditional and prior probabilities. On the other hand, BNs are used to estimate marginal probabilities conditional on observed data using Bayes' formula (equation 4.1).

Segment duration prediction is accomplished via *Bayesian inference*; Bayesian inference means calculating the probability of some variables given the values of some other variables (calculating marginals). In particular, we use the joint probability distribution $P(\mathbf{X})$ to calculate the probability of a segment taking on a particular duration given the observations of the linguistic factors that are known to affect its duration.

Once the BN structure and the joint probability distribution are specified, one can perform an inference on the network by summing out irrelevant variables. As was discussed in Section 4.1, direct marginalisation in the BN domain can be computationally infeasible. Instead, a Bayesian network is converted into a junction tree, with belief potentials being defined over a set of cliques; and an inference is performed on the junction tree.

4.4.1 Conditional Gaussian (CG) networks

A problem domain variable can either represent qualitative information, taking on discrete values, or it can represent quantitative information, therefore taking on continuous values. Depending on whether all the variables in the network are discrete, continuous, or a mixture of both, BNs can be categorised as discrete, continuous, and hybrid networks. In order to efficiently exploit the local structure of a hybrid BN, its problem domain set $\mathbf{X}$ is

divided into two subsets: a subset of discrete variables $\mathbf{I}$ and a subset of continuous variables $\mathbf{Y}$, i.e. $\mathbf{X} = \mathbf{I} \cup \mathbf{Y}$ (see Cowell et al. (1999) for details).

For segment duration prediction we use a special kind of a hybrid Bayesian network, namely a Conditional Gaussian (CG) network. We say that the variables $X_j \in \mathbf{X}; j = 1, \cdots, n$ of a hybrid BN follows *conditional Gaussian* (CG) distribution, if the BN's continuous variables follow a multivariate Gaussian distribution given the values of the discrete variables[4]. The

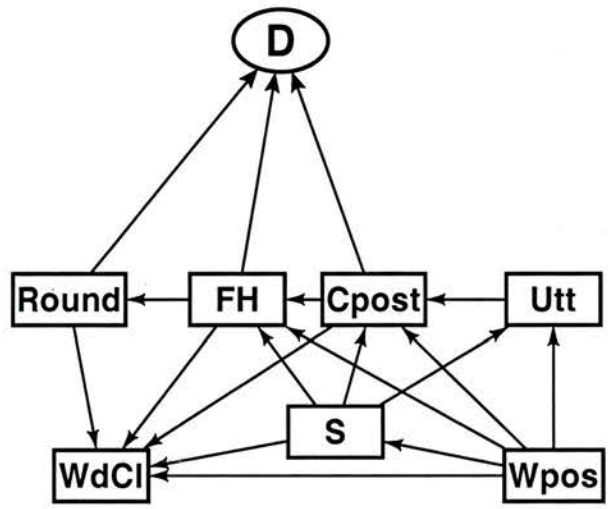

Figure 4.7: The *FH-compound* network learnt by the K2 algorithm, with vowel durations being uniformly discretised. The VBN2-8 model.

problem domain of a durational CG network consists of discrete nodes that represent linguistic factors that affect segment duration, and a continuous node (further on denoted D) representing a segment's duration. For example, the BN of size 8 for predicting a vowel duration has a variable set $\mathbf{X}$ ={ *Wpost, S, Utt,Cpos, FH, Rnd, Wd, D* }. As can be seen from Figure B.2

[4]In the analysis to follow we will use the terms *hybrid* and *Conditional Gaussian* (CG) interchangeably. In general, the terms are not equivalent, the latter being a special case of the former.

the subset of discrete nodes $\mathbf{I}$ consists of 7 linguistic factors $\mathbf{I} = \{$ *Wpost, S, Utt, Cpos, FH, Rnd, Wd* $\}$ where *Wpost* is the within-word position factor, *S* is the lexical stress of the word, *Utt* is the within-utterance position of the word containing a target vowel, *Cpos* is the following segment identity, *FH* is the vowel identity factor based on frontness and height of the vowel, *Rnd* is the roundness, and *Wd* is the word class of the word containing the target vowel. The set of continuous variables $\mathbf{Y}$ consists of one node D, the segment's duration: $\mathbf{Y} = \{ D \}$.

Let $\mathbf{x} = (\mathbf{i}, \mathbf{y})$ be an instantiation of the variables $\mathbf{X}$ in a CG network, where $\mathbf{i}$ is the instantiation of the discrete variables from the subset $\mathbf{I}$, and $\mathbf{y}$ is the instantiation of the continuous variables from the subset $\mathbf{Y}$. Given a CG Bayesian network, its CG probability density function (pdf) is given by:

$$p(\mathbf{y}|\mathbf{i}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma(\mathbf{i})|}} \, exp\{-\frac{1}{2}(\mathbf{y} - \vec{\mu}(\mathbf{i}))^T \, \Sigma(\mathbf{i})^{-1} \, (\mathbf{y} - \vec{\mu}(\mathbf{i}))\}, \tag{4.8}$$

where n is the cardinality of the set $\mathbf{Y}$; $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ are the instantiations of the continuous variables given the instantiation of discrete parents $\mathbf{i}$ $\vec{\mu}(\mathbf{i})$ and $\Sigma(\mathbf{i})$ are the mean vector and covariance matrix of the multivariate Gaussian distribution given the values of the discrete nodes $\mathbf{i}$; here the covariance matrix Σ is positive definite. When $n = 1$, the subset of continuous variables consists of just the duration variable: $\mathbf{Y} = D$, We assume that duration D follows a 1-dimensional CG distribution, its *probability density function* (*pdf*) is written as:

$$p(d|\mathbf{i}) = \frac{1}{\sqrt{(2\pi)}\sigma(\mathbf{i})} \, exp\{-\frac{(d - \mu(\mathbf{i}))^2}{2\sigma^2(\mathbf{i})}\}, \tag{4.9}$$

where for each instantiation of the discrete variables $\mathbf{i}$ the value d is the duration of a segment, $\mu(\mathbf{i})$ and $\sigma^2(\mathbf{i})$ are its conditional mean and variance, accordingly.

4.4.2 CG potentials

We generalise a CG distribution (equation 4.8) to a CG potential , which is a function $\phi(\mathbf{x})$ (not necessarily normalised to 1) defined as follows:

$$\phi(\mathbf{x}) = \phi(\mathbf{y}, \mathbf{i}) = \chi(\mathbf{i}) \exp\{g(\mathbf{i}) + \mathbf{h}(\mathbf{i})^T \mathbf{y} - \frac{1}{2}\mathbf{y}^T K(\mathbf{i})\mathbf{y}\} \tag{4.10}$$

where $\chi(\mathbf{i}) \in \{0, 1\}$ is an indicator function which is equal to 1 whenever $P(X_\mathbf{I} = \mathbf{i}) > 0$; and $K(\mathbf{i})$ is symmetric, not necessarily invertible. The triple $(g, \mathbf{h}, K)$ is called the *canonical characteristics* of a potential $\phi_\mathbf{x}$. If $\chi(\mathbf{i}) = 1$ and $K(\mathbf{i})$ is positive definite, then the moment characteristics $(p(\mathbf{i}), \vec{\mu}(\mathbf{i}), \Sigma(\mathbf{i}))$ are defined as well:

$$\begin{aligned} \Sigma(\mathbf{i}) &= K(\mathbf{i})^{-1} \\ \mu(\mathbf{i}) &= \Sigma(\mathbf{i})\mathbf{h}(\mathbf{i}) \\ p(\mathbf{i}) &= \chi(\mathbf{i})(2\pi)^{d/2}|K(\mathbf{i})|^{-\frac{1}{2}} \exp\{g(\mathbf{i}) + \frac{1}{2}\mathbf{h}(\mathbf{i})^T K(\mathbf{i})^{-1}\mathbf{h}(\mathbf{i})\} \end{aligned} \tag{4.11}$$

There are operations of extension, multiplication, division, and marginalisation defined for CG potentials (Cowell et al. (1999)).

Let ϕ be a CG potential defined on $\mathcal{U} = (\mathcal{I} \times \mathcal{Y})$, where $\mathcal{I}$ is the domain of the discrete variables $\mathbf{I}$, $\mathcal{Y}$ is the domain of the continuous variables $\mathbf{Y}$, and $\mathcal{U}$ is set to be the Cartesian product of the domains $\mathcal{I}$ and $\mathcal{Y}$. (Cartesian product is denoted by $\times$.) The potential is **extended** to $\bar{\phi}$ on $\mathcal{W} = (\mathcal{I} \times \mathcal{J}) \times (\mathcal{Y} \times \mathcal{Z})$ by setting $\bar{\phi}(\mathbf{i}, \mathbf{j}, \mathbf{y}, \mathbf{z}) = \phi(\mathbf{i}, \mathbf{y})$. In practice, when a potential defined on a set of variables is extended to a larger set of variables, the vector $\mathbf{h}$ and matrix K are enlarged to the appropriate size and the corresponding values are set to zeros.

A potential ϕ_1 defined on $\mathcal{U}$ is **multiplied** by a potential ϕ_2 defined on $\mathcal{W}$, giving a potential (called *product*) $\phi_1 \star \phi_2$ defined on $\mathcal{U} \cup \mathcal{W}$:

$$(\phi_1 \star \phi_2)(\mathbf{x}) = \phi_1(\mathbf{x}) \star \phi_2(\mathbf{x})$$

with $\phi_1(\mathbf{x})$ and $\phi_2(\mathbf{x})$ on the right hand side being extended to $\mathcal{U} \cup \mathcal{W}$. Multiplication of potentials is equivalent to the addition of the components

of the canonical characteristics:

$$(g_1, \mathbf{h_1}, K_1) \times (g_2, \mathbf{h_2}, K_2) = (g_1 + g_2, \mathbf{h_1} + \mathbf{h_2}, K_1 + K_2)$$

A potential ϕ_1 is **divided** by potential ϕ_2 like so:

$$\phi_1/\phi_2 = \begin{cases} (\phi_1/\phi_2)(\mathbf{x}), & \text{if} \quad \phi_2 \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Likewise, the operation of division expressed in canonical characteristics:

$$(g_1, \mathbf{h_1}, K_1)/(g_2, \mathbf{h_2}, K_2) = (g_1 - g_2, \mathbf{h_1} - \mathbf{h_2}, K_1 - K_2)$$

The potential $\bar{\phi}$ defined on $\mathbf{U}$ is called a **marginal** of the potential ϕ defined on $\mathbf{W}$, if it is derived by marginalising the continuous variables followed by marginalising the discrete variables.

It should be pointed out that sum of two CG potentials does not belong to the CG conjugate family, i.e. the sum of two CG potentials is a mixture of CG potentials. For that reason, marginalisation of CG potentials is handled differently from purely discrete or continuous cases. The operation of marginalisation proceeds in two steps as described in Cowell et al. (1999). First, the marginalisation over the continuous variables is performed; this is simply an integration. Second, the marginalisation (summation) of discrete variables is done. When marginalising over discrete variables, there are two cases to be distinguished.

1. Neither $\mathbf{h}$ nor K depend on marginalised variables, this is called *strong marginalisation.* If $\mathbf{h}(\mathbf{i},\mathbf{j})$ and $K(\mathbf{i},\mathbf{j})$ are independent of $\mathbf{j}$, i.e., $\mathbf{h}(\mathbf{i},\mathbf{j}) = h(\mathbf{i})$ and $K(\mathbf{i},\mathbf{j}) = K(\mathbf{i})$; the marginal $\bar{\phi}$ of ϕ over $\mathbf{j}$ could be written:

$$\bar{\phi} = \exp\{\mathbf{h}^T(\mathbf{i})\mathbf{y} - \frac{1}{2}\mathbf{y}^T K(\mathbf{i})\} \sum_{\mathbf{j}} \chi(\mathbf{i},\mathbf{j}) \exp\{g(\mathbf{i},\mathbf{j})\}$$

.

2. $\mathbf{h}$ and K depend on marginalised variables; this is called *weak marginalisation.* The marginal $\bar{\phi}(\mathbf{i})$ is defined with the *moment characteristics*

$\tilde{p}(\mathbf{i}), \tilde{\mu}(\mathbf{i}), \tilde{\Sigma}(\mathbf{i})$, where

$$\begin{aligned} \tilde{p}(\mathbf{i}) &= \sum_{\mathbf{j}} p(\mathbf{i},\mathbf{j}) \\ \tilde{\mu}(\mathbf{i}) &= \sum_{\mathbf{j}} \mu(\mathbf{i},\mathbf{j}) \frac{p(\mathbf{i},\mathbf{j})}{\tilde{p}(\mathbf{i})} \\ \tilde{\Sigma}(\mathbf{i}) &= \sum_{\mathbf{j}} \Sigma(\mathbf{i},\mathbf{j}) \frac{p(\mathbf{i},\mathbf{j})}{\tilde{p}(\mathbf{i})} + \sum_{\mathbf{j}} (\mu(\mathbf{i},\mathbf{j}) - \tilde{\mu}(\mathbf{i}))^T (\mu(\mathbf{i},\mathbf{j}) - \tilde{\mu}(\mathbf{i})) \frac{p(\mathbf{i},\mathbf{j})}{\tilde{p}(\mathbf{i})} \end{aligned} \tag{4.12}$$

4.5 Inference on the junction tree

Having constructed the junction tree according to the steps described in Section 4.3 we are ready to perform the probabilistic inference on the junction tree: we can compute the probability distribution of any variable of the original problem domain $\mathbf{X}$ given the observed variables.

4.5.1 Initialisation

First, the junction tree built has to be initialised so that the JPD $P(\mathbf{X})$ is preserved (equation 4.6). For each clique $\mathbf{C_k} \in \mathbf{C}$ ($k = 1, \cdots, m$; m is the number of cliques) and separator $\mathbf{S_l} \in \mathbf{S}$ ($l = 1, \cdots, m-1$) the potentials are initialised to 1:

$$\phi_{\mathbf{C_k}} \leftarrow 1 \qquad \phi_{\mathbf{S_l}} \leftarrow 1$$

For each variable $X_j \in \mathbf{X}$ ($j = 1, \cdots, n$) assign its family $F_{X_j} = \{X_j, \mathbf{Pa}(X_j)\}$ to any clique $\mathbf{C_k}$ that contains F_{X_j}. Multiply $\phi_{\mathbf{C}}$ by $P(X_j|\mathbf{Pa}(X_j))$:

$$\phi_{\mathbf{C_k}} \leftarrow \phi_{\mathbf{C_k}} P(x_j|\mathbf{pa}(X_j))$$

After the initialisation, Equation 4.6 is satisfied:

$$\frac{\prod_{k=1}^{m} \phi_{\mathbf{C_k}}}{\prod_{l=1}^{m-1} \phi_{\mathbf{S_l}}} = \frac{\prod_{j=1}^{n} P(X_j|\mathbf{Pa}(X_j))}{1} = P(\mathbf{X}),$$

where n is the number of variables, m is the number of cliques in the junction tree; and $\phi_{\mathbf{C_k}}$ and $\phi_{\mathbf{S_l}}$ are clique and separator potentials, respectively.

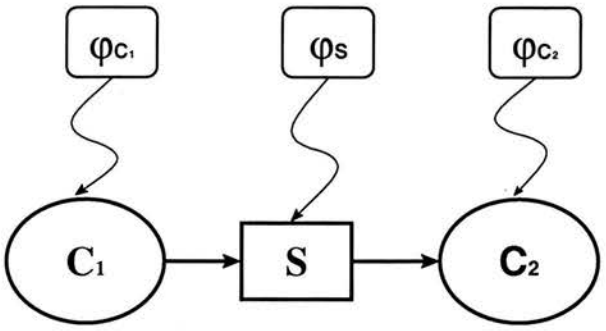

Figure 4.8: Single message pass example

4.5.2 Global propagation

Thus far, the initialised junction tree is locally inconsistent due to the arbitrarily assigned initial potentials' values. Hence, the condition of local consistency (equation 4.5) is not satisfied yet. In order to make the junction tree locally consistent, an ordered series of local procedures (called *message passes*) are performed on the junction tree potentials (see Huang & Darwiche (1996)).

The message passing algorithm consists of two passes termed COLLECT EVIDENCE (or collect-to-root) and DISTRIBUTE EVIDENCE (distribute -from-root).

1. COLLECT EVIDENCE (Collect-to-root pass). A clique $\mathbf{C_k} \in \mathbf{C}$ is called COLLECT EVIDENCE from a neighbour $\mathbf{C_p}$; then $\mathbf{C_k}$ calls COLLECT EVIDENCE in all (but $\mathbf{C_p}$) its other neighbours. When they have finished their COLLECT EVIDENCE, a message is sent from them towards $\mathbf{C_k}$.

   In the COLLECT EVIDENCE pass each clique sends a message to the strong root[5] after receiving messages from its children in post-

[5] A node R of the junction tree $\mathcal{T}$ is a strong root, if any pair A, B of neighbours on the tree with A closer to R than B satisfies: $(B \backslash A) \subseteq \mathbf{Y}$ or $(B \cap A) \subseteq \mathbf{I}$, where $\mathbf{Y}$ and $\mathbf{I}$

order[6]. After COLLECT EVIDENCE has finished, the root will have the information about every clique and separator in the tree.

2. DISTRIBUTE EVIDENCE (Distribute-from-root). When DISTRIBUTE EVIDENCE is called in $\mathbf{C_k}$ from a neighbour $\mathbf{C_p}$, then $\mathbf{C_p}$ passes a message to $\mathbf{C_k}$ and calls DISTRIBUTE EVIDENCE in all (but $\mathbf{C_k}$) its other neighbours.

   In the DISTRIBUTE EVIDENCE pass, the root sends messages to its children in pre-order[7]. After DISTRIBUTE EVIDENCE has terminated, each clique has sent the information encoded in its belief potential to every other clique in the tree. Therefore, the junction tree will be locally consistent.

Let us consider a single message pass between adjacent cliques $\mathbf{C_1}$, $\mathbf{C_2}$ and separator $\mathbf{S}$ (Figure 4.8), with corresponding belief potentials being $\phi_{\mathbf{C_1}}$, $\phi_{\mathbf{C_2}}$ and $\phi_{\mathbf{S}}$. We assume that $\mathbf{C_1}$ sends a message to $\mathbf{C_2}$; the former is called the *source*, and the latter is called the *sink*. There are two operations that are used to restore the local consistency in the junction tree: *projection* and *absorption* operations. When the clique $\mathbf{C_1}$ sends a message to its neighbour $\mathbf{C_2}$, the separator potential is updated by the *projection* operation (i.e. saving the old potential):

$$\phi_{\mathbf{S}^{old}} \leftarrow \phi_{\mathbf{S}} \tag{4.13}$$

And a new potential is assigned to the separator $\mathbf{S}$ by marginalising the source's potential $\phi_{\mathbf{C_1}}$:

$$\phi_{\mathbf{S}} \leftarrow \sum_{\mathbf{C_1} \setminus \mathbf{S}} \phi_{\mathbf{C_1}}$$

Following the projection, the sink $\phi_{\mathbf{C_2}}$ *absorbs* from the source $\phi_{\mathbf{C_1}}$:

$$\phi_{\mathbf{C_2}} \leftarrow \phi_{\mathbf{C_2}} \frac{\phi_{\mathbf{S}}}{\phi_{\mathbf{S}^{\mathbf{old}}}} \tag{4.14}$$

---

are the subsets of continuous and discrete nodes, respectively.

[6]Children following the parents.

[7]Children going before parents

Jensen (1996) shows that if for any instantiation $\mathbf{s} \in \mathbf{S}$ the potential $\phi_{\mathbf{S}}(\mathbf{s}) = 0$, then the old potential $\phi_{\mathbf{S}^{old}}(\mathbf{S})$ is set to 0, i.e. $\phi_{\mathbf{S}^{old}}(\mathbf{S}) = 0$; when this happens, we assign $0/0 = 0$.

The projection and absorption operations preserve the joint probability distribution $P(\mathbf{X})$:

$$\left(\frac{\prod_k^m \phi_{\mathbf{C_k}}}{\prod_l^{m-1} \phi_{\mathbf{S_l}}}\right) \frac{\phi_{\mathbf{S}^{old}}}{\phi_{\mathbf{S}}} \frac{\phi_{\mathbf{C_2}}}{\phi_{\mathbf{C_2^{old}}}} = \left(\frac{\prod_k \phi_{\mathbf{C_k}}}{\prod_l \phi_{\mathbf{S_l}}}\right) \frac{\phi_{\mathbf{S}^{old}}}{\phi_{\mathbf{S}}} \frac{\phi_{\mathbf{C_2}^{old}} \frac{\phi_{\mathbf{S}}}{\phi_{\mathbf{S}}^{old}}}{\phi_{\mathbf{C_2^{old}}}} = P(\mathbf{X})$$

Single message passes are coordinated in such a way as to maintain local consistency in a junction tree. During the COLLECT-EVIDENCE phase, each cluster sends a message to its neighbour in the direction of the clique $\mathbf{C}$, beginning with the cliques farthest from the clique $\mathbf{C}$. During the DISTRIBUTE-EVIDENCE phase each clique passes messages to its neighbours away from the direction of the clique $\mathbf{C}$, beginning with the clique $\mathbf{C}$ itself. Each clique only sends a message to a neighbour when it received messages from all of its other neighbours. After the COLLECT-EVIDENCE and DISTRIBUTE-EVIDENCE finished there are $2(m-1)$ messages sent, and the junction tree is locally consistent.

Now one can compute the probability distribution $P(X_j)$ of any variable $X_j \in \mathbf{X}$. First, we identify the clique $\mathbf{C}$ (or separator $\mathbf{S}$) to which X_j belongs, i.e. $X_j \in \mathbf{C}$ ($X_j \in \mathbf{S}$). Then we compute $P(X_j)$ by marginalising $\phi_{\mathbf{C}}$ ($\phi_{\mathbf{S}}$) as in equation 4.7 repeated here for convenience:

$$P(X_j) = \sum_{\mathbf{C}\setminus\{X_j\}} \phi_{\mathbf{C}}$$
$$(P(X_j) = \sum_{\mathbf{S}\setminus\{X_j\}} \phi_{\mathbf{S}})$$

4.6 Learning Bayesian network structure

The problem of learning in Bayesian Networks presupposes the availability of data. Data is defined as a collection of *feature vectors*, i.e. $\mathcal{D}_M = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_M}\}$, where M is the total number of feature vectors in the database. Each feature vector $\mathbf{x_j} = (x_{1j}, \ldots, x_{nj})$ (where n is the number of

variables in the BN, and $j = 1, \cdots, M$) is a vector of variables, with each variable taking on a particular value from the finite set of values (in the case of a discrete variable) or the set of real numbers (in the case of a continuous variable).

There are two different types of learning from data in Bayesian Networks. On the one hand, a qualitative graphical representation of a network can be learnt; this is called *structure learning*. On the other hand, a quantitative specification, i.e., conditional probability distributions (CPDs) for each variable in the network, can be learnt; this is called *parameter learning*. The learning could be done either for fully observed data or partially observed data (when some of the variables are hidden or missing).

The most common approach to network structure learning is to apply some heuristic search techniques to search through the hypothesis space of possible network structures and evaluating a scoring metric (function) for each candidate network. Subsequently, the network with the highest score is selected. To this end, there are two common scoring functions used in network structure learning algorithms: *Bayesian measure* and *Minimum Description Length* (MDL). We will review the structure learning algorithm based on the *Bayesian measure* here, since we used it to learn the structure of the durational BNs. Structure learning using the MDL scoring function is covered in much detail in Lam & Bachus (1994) and Bouckaert (1994).

4.6.1 Bayesian measure

A Bayesian approach to structure learning comes down to maximising the probability of a network structure G given the database $\mathcal{D}_M$, $P(G|\mathcal{D}_M)$. In other words, given a set of candidate network structures the goal is to find a network structure G' that produces a maximum value of $P(G'|\mathcal{D}_M)$ for a database of cases $\mathcal{D}_M$. Given two network structures G_1 and G_2, we can compare the probabilities:

$$\frac{P(G_1|\mathcal{D}_M)}{P(G_2|\mathcal{D}_M)} = \frac{\frac{P(G_1,\mathcal{D}_M)}{P(\mathcal{D}_M)}}{\frac{P(G_2,\mathcal{D}_M)}{P(\mathcal{D}_M)}} = \frac{P(G_1,\mathcal{D}_M)}{P(G_2,\mathcal{D}_M)}$$

Since for all BNs the probability $P(\mathcal{D}_M)$ is the same, it suffices to calculate the joint probability $P(G, \mathcal{D}_M)$ for all networks G. The joint probability $P(G, \mathcal{D}_M)$, also called the *Bayesian measure* (see Cooper & Herskovits (1992) for details), is given by:

$$P(G, \mathcal{D}_M) = P(G) \prod_{j=1}^{n} \prod_{k=1}^{q_j} \frac{(r_j - 1)!}{(N_{jk} + r_j - 1)!} \cdot \prod_{l=1}^{r_j} N_{jkl}! \qquad (4.15)$$

where r_j is the number of values the node X_j of a network G can take on, and $q_j = \prod_{p \in \mathbf{Pa}(X_j)} r_p$ is the number of values that X_j's parents $\mathbf{Pa}(X_j)$ can take on; N_{jkl} is the number of times the event $(X_j = l, \mathbf{Pa}(X_j) = k)$ occurs in the database $\mathcal{D}_M$, and $N_{jk} = \sum_{l=1}^{r_j} N_{jkl}$; $P(G)$ is the prior probability of network structure G. Its value is usually elicited from the experts. Otherwise, with no prior information, $P(G)$ is assumed to be uniformly distributed; hence, the term $P(G)$ can be neglected when the two structures are compared.

4.6.2 Bayesian measure-based search approach

INPUT: ordered list of variables $\{X_1, \cdots, X_n\} \in U$

OUTPUT: G defined by new parent sets $\{\pi_{1,new}, \cdots, \pi_{n,new}\}$

Let the variables of U be ordered $X_1, X_2, \cdots, X_n$

for $i = 1, \cdots, n$ **do** $\pi_{i,old} \leftarrow \emptyset$

for $i = 2, \cdots, n$ **do**

  repeat

    $\pi_{i,old} \leftarrow \pi_{i,new}$

    Let G be defined by $\pi_{1,old}, \cdots, \pi_{n,old}$

    $z \leftarrow argmax_y \{\frac{P(G_y, \mathcal{D}_M)}{P(G, \mathcal{D}_M)} | y \in \{X_1, X_2, \cdots, X_{i-1}\} \backslash \pi_{i,old}$, where

      G_y is G but with $\pi_i = \pi_{i,old} \cup \{y\}$ }

    if $\frac{P(G_y, \mathcal{D}_M)}{P(G, \mathcal{D}_M)} > 1$ **then** $\pi_{i,new} \leftarrow \pi_{i,old} \cup \{z\}$

  until $\pi_{i,new} = \pi_{i,old}$ **or** $|\pi_{i,new}| = i - 1$

Table 4.1: BN structure learning algorithm K2

The number of all possible networks of n nodes is huge; it is given by

this recursive formula taken from Cowell et al. (1999):

$$N(n) = \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{(n-1)!i!} 2^{i(n-i)} N(n-i) \tag{4.16}$$

As can be seen from Equation 4.16, the number of all possible graphs on n nodes is hyperexponential in the number of nodes n. Therefore, direct exhaustive search is computationally prohibitive. Consequently, we have to resort to some heuristic approaches to find the best graph structure G given the data.

Based on the Bayesian measure, Cooper & Herskovits (1992) developed an algorithm, called K2, for learning a network structure from discrete-valued data. In the algorithm, it is assumed that the ordering of the nodes is given. This assumption reduces the computation complexity to $\mathcal{O}(2^{n^2})$. As this number is still huge, Cooper & Herskovits (1992) came up with a greedy heuristic search that cuts down the time complexity even further, bringing it down to $\mathcal{O}(n^3)$. The K2 algorithm starts with an empty parent set for each variable X_j; $j = 1, \ldots, n$. It successively adds the variable' from the set $\{X_1, \ldots, X_{j-1}\}$ that maximally improves the Bayesian measure $P(G, \mathcal{D}_M)$. The pseudocode of the K2 search algorithm is presented in Figure 4.1.

It must be pointed out that the node ordering greatly affects the quality of the network structure learnt. Therefore, it is essential to provide a good node ordering to guarantee a good K2 performance.

4.7 Learning Bayesian network parameters

There are two main approaches to parameter learning. One is Maximum Likelihood Estimation (MLE), which is a classical statistical approach, whereby a single best parameter estimate given the data is learnt. Another is Bayesian parameter learning whereby the posterior distribution of the parameters is learnt given the prior distribution of the parameters and the data. We will briefly review both approaches below.

Suppose our Bayesian network is described by some probability density

function that is dependent on the set of parameters $\mathbf{\Theta}$. In addition, suppose we have a training database $\mathcal{D}_M = \{\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_M}\}$ of M *feature vectors*, or *training cases* drawn from this distribution. We assume that the training data are *iid*, i.e., *independently and identically distributed.* First, this implies that each feature vector $\mathbf{x_{m_1}}$ is generated independently of every other feature vector $\mathbf{x_{m_2}}$ $(m_1, m_2 = 1, \cdots, M;\ m_1 \neq m_2)$. Second, it assumes that all feature vectors are generated from the same distribution. The *pdf* for the data $\mathcal{D}_M$ is written as:

$$p(\mathcal{D}_\mathcal{M}|\mathbf{\Theta}) = \prod_{m=1}^{M} p(\mathbf{x}_m|\mathbf{\Theta}) = \mathcal{L}(\mathbf{\Theta}|\mathcal{D}_\mathcal{M}) \tag{4.17}$$

where $\mathbf{\Theta} = (\theta_\mathbf{1}, \theta_\mathbf{2}, \cdots, \theta_\mathbf{M})$ is the vector of BN parameters, with $\theta_\mathbf{m}$ being a vector of parameters for the feature vector $\mathbf{x}_m$. This function $\mathcal{L}(\mathbf{\Theta}|\mathcal{D}_\mathcal{M})$ is called the *likelihood function*, or just *likelihood.* It is assumed to be a function of the Bayesian parameters $\mathbf{\Theta}$ when the data $\mathcal{D}_M$ is fixed.

For parameter estimation, it is convenient to use the log-likelihood function written as:

$$L = log\ \mathcal{L} = \sum_{m=1}^{M} log\ p(\mathbf{x}_m|\mathbf{\Theta}) \tag{4.18}$$

4.7.1 Full observability: MLE parameter estimation

Suppose all the variables in the database $\mathcal{D}_M$ are observed (full observability). Hence, we can apply the MLE approach to calculate the optimal estimates of the parameters Θ given the database of training cases $\mathcal{D}_M$ by maximising the log-likelihood function (Equation 4.18). Let us assume all the variables in the network are continuous and described by a multivariate Gaussian distribution with the *pdf* written as:

$$p(\mathbf{x}|\mathbf{\Theta}) = \frac{1}{\sqrt{(2\pi)^n\ |\Sigma|}}\ exp\{-\frac{1}{2}(\mathbf{x} - \vec{\mu})^T\ \Sigma^{-1}\ (\mathbf{x} - \vec{\mu})\}, \tag{4.19}$$

where $\mathbf{\Theta} = (\vec{\mu}, \Sigma)$ are the parameters, with the mean $\vec{\mu}$ being a n-dimensional vector, and Σ being $n \times n$ covariance matrix; $|\Sigma|$ is the determinant of Σ.

The log-likelihood function is therefore, written as:

$$L(\Theta) = \sum_{m=1}^{M} \frac{1}{\sqrt{(2\pi)^n \ |\Sigma|}} \ exp\{-\frac{1}{2}(\mathbf{x_m} - \vec{\mu})^T \ \Sigma^{-1} \ (\mathbf{x_m} - \vec{\mu})\}, \qquad (4.20)$$

where $\mathbf{x_m}$ is the m-th feature vector of the database $\mathcal{D}_M$, $\vec{\mu}$ and Σ are defined similar to the Equation 4.19. Then the MLE estimates of the parameters $\Theta = (\vec{\mu}, \Sigma)$ can be found by analytical differentiation of Equation 4.20 (see, for example, references mentioned in Bishop (1998)):

$$\hat{\mu} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{x_m} \qquad (4.21)$$

$$\hat{\Sigma} = \frac{1}{M} \sum_{m=1}^{M} (\mathbf{x_m} - \hat{\mu})(\mathbf{x_m} - \hat{\mu})^T \qquad (4.22)$$

Now let us assume all the variables in the network are discrete. Therefore, we assume each variable $X_j \in \mathbf{X}$ has r_j possible values $x_j^1, \cdots, x_j^{r_j}$; and the parents of X_j, $\mathbf{Pa}(X_j)$ can take on $q_j = \prod_{p \in \mathbf{Pa}(X_j)} r_p$ values, i.e. $\mathbf{pa}(X_j) = (pa_j^1, pa_2^1, \cdots, pa_j^{q_j})$. We assume each local distribution function is collection of multinomial distributions, one distribution for each configuration of the parents $\mathbf{pa}_j^k$, $k = 1, \cdots, q_j$. Following Heckerman (1995), we denote the parameters as:

$$\theta_{jkl} = P(X_j = l | \ \mathbf{Pa}(X_j) = k, \theta_\mathbf{j}), \qquad (4.23)$$

where $\theta_{jkl} > 0$ and $\sum_l \theta_{jkl} = 1$ are the entries in the conditional probability table (CPT) that are specified for each variable X_j; we can calculate the parameter θ_{jk1} as: $\theta_{jk1} = 1 - \sum_{l=2}^{r_j} \theta_{jkl}$. For the variable X_j there exist the parameters $\theta_\mathbf{j} = ((\theta_{jkl})_{l=2}^{r_j})_{k=1}^{q_j}$. For convenience, we also define a vector of parameters:

$$\theta_{jk} = (\theta_{jk2}, \ldots, \theta_{jkr_j})$$

for all j and k. We re-write the log-likelihood function of Equation 4.18 as:

$$L(\Theta) = \sum_{j=1}^{n} \sum_{m=1}^{M} log \prod_{k,l} \theta_{jkl}^{\chi_{jklm}} = \sum_{j} \sum_{m} \sum_{k,l} \chi_{jklm} \ log\theta_{jkl} = \sum_{jkl} N_{jkl} \ log\theta_{jkl} \qquad (4.24)$$

where $\chi_{jklm} = \chi(X_j = l,\ \mathbf{Pa}(X_j) = k|\mathbf{x}_m)$ is an indicator function that is equal to 1 if the event $(X_j = l,\ \mathbf{Pa}(X_j) = k)$ occurs in the training case $\mathbf{x}_m$. Therefore, $N_{jkl} = \sum_m \chi(X_j = l,\ \mathbf{Pa}(X_j) = k|\mathbf{x}_m)$ is defined to be the number of times the event $(X_j = l,\ \mathbf{Pa}(X_j) = k)$ occurs in the database $\mathcal{D}_M$. Using a Lagrange multiplier technique described, for example, in Bishop (1998) the MLE estimate is given:

$$\widehat{\theta_{jkl}} = \frac{N_{jkl}}{\sum_p N_{jkp}} \tag{4.25}$$

4.7.2 Full observability: Bayesian parameter estimation

Suppose our prior uncertainty about the parameters $\mathbf{\Theta}$ is specified by the distribution $p(\mathbf{\Theta})$. Hence, the problem of Bayesian parameter learning states that given a random sample $\mathcal{D}_M$, one can compute the posterior distribution of the parameters $p(\mathbf{\Theta}|\ \mathcal{D}_M)$.

Let us consider the discrete case first. Suppose each variable $X_j \in \mathbf{X}$ is discrete and represented with a collection of unrestricted multinomial distributions as described in Section 4.7.1.

In order to perform Bayesian parameter estimation we make two assumptions. First, we assume the Bayesian parameters $\mathbf{\Theta}$ are *globally* independent, i.e., the likelihood $\mathcal{L}(\mathbf{\Theta}|\mathcal{D}_M)$ factorises into the product of the marginal distributions for each individual training case $\mathbf{x_m}$, i.e.,

$$\mathcal{L}(\mathbf{\Theta}|\mathcal{D}_M) = \prod_{m=1}^{M} \mathcal{L}(\theta_{\mathbf{m}}|\mathcal{D}_M)$$

where $\theta_{\mathbf{m}} = (\theta_{mjk}); (k = 1, \cdots, r_m)$ is the vector of parameters for the training case $\mathbf{x_m}$.

Second, we assume the Bayesian parameters $\mathbf{\Theta}$ are *locally* independent, i.e. for each variable X_j its local parameters factorise according to all possible instantiations q_j of the parents $Pa(X_j)$:

$$\theta_{\mathbf{j}} = \prod_{k=1}^{q_j} \theta_{jk}$$

Given the parameters Θ, the joint probability distribution of a Bayesian network (Equation 4.3) can be rewritten as follows:

$$P(X_1, X_2, ..., X_n|\Theta) = \prod_{j=1}^{n} P(X_j|\mathbf{Pa}(X_j),\ \theta_{\mathbf{j}})$$

where $\Theta = (\theta_{\mathbf{1}}, \cdots, \theta_{\mathbf{n}})$ is the vector of parameters

The local parameter independence assumption states that for each variable X_j the parameters of the parents $\mathbf{Pa}(X_j)$ are independent, i.e., $P(\theta_j) = \prod_{k=1}^{q_j} \theta_{jk}$, where $\theta_{jk} = \{\theta_{jkl},\ l = 1, \cdots, r_j\}$ are the parameters for the k'th instantiation of the X_j's parents.

In order to simplify analysis, we assume that the functional form of the prior distribution is a *conjugate prior*[8]. For example, the family of Dirichlet distributions is a conjugate for the multinomial sampling.

Given global and local independence, each CPD $P(X_j|\ \mathbf{Pa}(X_j) = k) = \theta_{jk}$ is a multinomial random variable with r_j possible values. We assume that each vector θ_{jk} has a Dirichlet prior defined as:

$$P(\theta_{jk}|\alpha_{jk}) = \prod_{l=1}^{r_j} \theta_{jkl}^{\alpha_{jkl}-1} \times \frac{1}{B(\alpha_{jk1}, \ldots, \alpha_{jkr_j})} \tag{4.26}$$

The normalising constant $B(\cdot)$ is the r_j-dimensional Beta function

$$B(\alpha_1, \cdots, \alpha_r) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_{k=1}^{r} \Gamma(\alpha_k)}$$

where $\Gamma(\cdot)$ is the Gamma function: for $n > 0 : \Gamma(n) = (n-1)!$. The hyperparameters α_{jkl} have simple interpretation of pseudo counts. The value $\alpha_{jkl} - 1$ corresponds to the number of imaginary counts of event $(X_j = l,\ \mathbf{Pa}(X_j) = k)$ that has occurred in some virtual database. After seeing N_{jkl} cases of event $(X_j = l,\ \mathbf{Pa}(X_j) = k)$ in the database $\mathcal{D}_M$ the parameter posterior becomes:

$$\theta_{jk}|\mathcal{D}_M = \prod_{l=1}^{r_j} \theta_{jkl}^{\alpha_{jkl}-1+N_{jkl}} \times \frac{1}{B(\alpha_{jk1} + N_{jk1}, \ldots, \alpha_{jkr_j+N_{jkr_j}})} \tag{4.27}$$

[8] A prior distribution is said to be conjugate when the posterior distribution is of the same functional form as the prior distribution

The posterior mean is:

$$E[\theta_{jkl}|\mathcal{D}_M] = \frac{\alpha_{jkl} + N_{jkl}}{\sum_{l=1}^{r_j} \alpha_{jkl} + N_{jkl}} \tag{4.28}$$

4.7.3 Partial observability: EM algorithm

The EM algorithm is used when the data is incomplete or has missing (or hidden) values. For the purposes of Bayesian duration prediction, we assume that the data may have some hidden values. For example, in the case of consonant duration prediction, the information about the syllabic position of a consonant may be ambiguous, which is the case for ambisyllabic consonants. In such a case, the variable describing its syllabic position would be considered hidden.

Let us divide the *complete data* $\mathcal{D}_M$ into the set of the *observed* data $\mathcal{D}_M^{obs}$ (which is the set of the observed variables $\mathbf{x}^{obs} \in \mathcal{D}_M^{obs}$ of the database $\mathcal{D}_M$), and the set of the *missing* (or *hidden*) data $\mathcal{D}_M^h$. The observed data is generated by some distribution. The joint density function of the complete data $\mathcal{D}_M$ is then written:

$$p(\mathcal{D}_M|\Theta) = p(\mathcal{D}_M^{obs}, \mathcal{D}_M^h|\Theta) = p(\mathbf{x}^{obs}, \mathbf{x}^h|\Theta)\ p(\mathbf{x}^{obs}|\Theta) \tag{4.29}$$

Consequently, the *complete* data likelihood function $\mathcal{L}(\mathcal{D}_M^{obs}, \mathcal{D}_M^h|\Theta)$ becomes a function of the hidden variables $\mathbf{x}^h \in \mathcal{D}_M^h$. The EM algorithm runs in two steps.

E-step: The expected values of the parameters Θ are estimated from the complete data log-likelihood with respect to the hidden data $\mathcal{D}_M^h$ given the observed data $\mathcal{D}_M^{obs}$ and the current parameter estimates $\Theta^{(i-1)}$. In order to do this, an auxiliary function is defined:

$$Q(\Theta, \Theta^{(i-1)}) = E[log\ p(\mathcal{D}_M^{obs}, \mathcal{D}_M^h|\Theta)|\ \mathcal{D}_M^{obs}, \Theta^{(i-1)}] \tag{4.30}$$

where $\Theta^{(i-1)}$ are the current parameter estimates that will be used evaluate Equation 4.30, and Θ are the new parameters that will be optimised to

increase Q. The right side of the Equation 4.30 can be re-written as follows.

$$E[log\ p(\mathcal{D}_M^{obs}, \mathcal{D}_M^h|\Theta)|\mathcal{D}_M^{obs}, \Theta^{(i-1)}] = \tag{4.31}$$

$$\int_{\mathbf{x}^h} log\ p(\mathcal{D}_M^{obs}, \mathbf{x}^h|\Theta)\ f(\mathbf{x}^h|\mathcal{D}_M^{obs}, \Theta^{(i-1)})d\mathbf{x}^h \tag{4.32}$$

where $f(\mathbf{x}^h|\mathcal{D}_M^{obs}, \Theta^{(i-1)})$ is the marginal distribution of the hidden data $\mathcal{D}_M^h$ given the observed data $\mathcal{D}_M^{obs}$ and the current parameters $\Theta^{(i-1)}$.

M-step: The expectation of the complete data log-likelihood is maximised:

$$\Theta^{(i)} = argmax_\Theta\ Q(\Theta, \Theta^{(i-1)}) \tag{4.33}$$

By iterating between the expectation and maximisation steps the algorithm is bound to converge, as was shown by Dempster, Laird & Rubin (1977).

Chapter 5

Bayesian Models: Vowels

In this chapter we present two classes of Bayesian models for predicting vowel duration, the main difference between the two being in vowel identity representation. In Section 5.1 we describe the method that we used for defining our Bayesian models. Then in Section 5.2 we present the class of models (*FHLR* models) that are based on a 4-feature vowel identity representation. In this section we also discuss the models' structure learning, training and results. In Section 5.3 we describe the class of networks based on a 2-feature vowel identity representation (*FH-compound* models). We further discuss the models' structure learning, training and results. In Section 5.4 we discuss our best *FHLR* models and *FH-compound* models. We conclude this chapter with a summary of the results in Section 5.5.

5.1 Method

We used the following procedure when performing experiments involving Bayesian prediction of segment duration.

1. Prepare data by transforming it into the format used by Bayesian prediction routines.

2. Select the set of linguistic factors that affect a vowel's (or consonant's) duration.

3. Perform Bayesian structure learning by applying the K2 structure learning algorithm described in Section 4.6.2 to the training data with discretised durations.

4. Select a set of unique Bayesian networks based on the unique networks' adjacency matrices. Identify the classes of networks with different durational variable parent sets (the networks within a class have the same durational variable parent set, being different otherwise). We will call such classes of networks *durational parent set equivalent* classes and discuss this in more detail in Section 5.2.3.

5. Apply the EM algorithm described in Section 4.7.3 to learn the parameters of each candidate network using the original, continuously-valued duration data.

6. Perform Bayesian inference of test set vowels' (consonants') durations based on the BN parameters learnt. Compare the performance of the Bayesian model to the baseline CART and SoP models.

5.1.1 Performance metrics

In order to test the performance of the Bayesian model and to further compare it to the baseline CART and SoP models, we used 2 metrics: *sample correlation coefficient* between observed and predicted values of duration and *Root Mean Squared Error* (RMS error) in milliseconds (ms). The sample correlation coefficient is defined as (see for example, Lee (1997)):

$$r = \frac{\sum_{m=1}^{M}(d_m^{obs} - \bar{d}^{obs})(d_i^{pred} - \bar{d}^{pred})}{\sqrt{\{\sum_{m=1}^{M}(d_m^{obs} - \bar{d}^{obs})^2\}\{\sum_{i=m}^{M}(d_m^{pred} - \bar{d}^{pred})^2\}}} \tag{5.1}$$

where M is the size of the test set, d_m^{obs} is an observed duration of a phone in the m-th feature vector, $\bar{d}^{obs}$ is the mean observed duration across the test set; d_m^{pred} is a predicted duration of a phone in the m-th feature vector, $\bar{d}^{pred}$ is the mean predicted duration across the test set. We will refer to the sample correlation coefficient as *correlation coefficient*, or just *correlation*.

For the RMS error we adopt a definition used in Bishop (1998):

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(d_m^{obs} - d_m^{pred})^2} \tag{5.2}$$

where M is the size of the test set, d_m^{obs} is the observed duration, and d_m^{pred} is the predicted duration in the m-th feature vector, respectively.

5.1.2 Testing the performance of Bayesian models

We performed the training of the Bayesian models under various experimental conditions. In order to compare different experimental conditions we used a paired t-test that allows us to check if there is a significant difference between the durations predicted by two Bayesian networks. The independent variables tested were the sample correlation coefficient r defined in Equation 5.1 and the *Root Mean Squared Error* RMSE defined in Equation 5.2. The null hypothesis H_0 is that there will be no difference between the BN models (trained under two different experimental conditions), i.e. the mean correlation (RMS error) difference is zero: $\bar{r} = 0$. The null hypothesis H_0 is tested against the two-sided non-null hypotheses H_1 ($\bar{r} > 0$) by calculating the t statistic:

$$t = \frac{\bar{r}}{\sqrt{s_r^2/n}} \tag{5.3}$$

where $\bar{r}$ is the mean correlation difference between two experimental conditions, n is the number of pairs (the number of segment classes), and s_r^2 is the sample variance of the mean correlation differences. We compared the calculated t value to a t-distribution with $n - 1$ degrees of freedom, which gives the probability of finding such a value of t by chance. If the probability p is lower than some threshold value ($p < 0.01$, one-tailed, unless otherwise stated), we reject H_0 in favour of H_1. Therefore, we can claim that one Bayesian model predicts vowel duration with significantly higher correlation (or lower RMS error) values than the other BN model.

5.2 *FHLR* networks

5.2.1 Selecting variables for Bayesian domain set

Here we present the full results of the pilot study which was briefly described in Goubanova (2003). We experimented with the networks defined on the domain set consisting of 10 variables: 9 discrete variables for the linguistic factors affecting vowel duration and one continuous variable for vowel duration. We chose the following variables: the within-word position *Wpos*, the syllabic stress *S*, the within-utterance position *Utt*, the following segment identity *Cpos*, the class of the word containing the target vowel *Wd* discrete variables, and the continuous-valued duration variable *D*. In addition, the vowel identity was represented as 4 variables based on phonological distinctive features of frontness *Front*, height *Height*, length *Length*, and roundness *Rnd*. The variable names and example values are shown in Tables 3.1, 3.3 of Section 3.1 on pages 21, 23. The variables' domain is written as:
$\mathbf{U_{10}}$ ={*Front, Height, Length, Rnd, Wpos, S, Utt, Cpos, Wd, D*} We will further refer to the networks based on this domain set as *FHLR* networks.

5.2.2 Learning network structure

We applied the K2 structure learning algorithm (Section 4.6.2) to the data, with duration values being uniformly discretised. We discretised the z-scores of the duration values by assigning them to evenly spaced bins. We chose 9 levels of discretisation (from 2 to 9 bins). The example of a z-scores curve divided into 5 bins is shown in Figure 5.1. The discretisation procedure resulted in 9 discretised training sets per voice, with the total of $3 \times 9 = 27$ discretised training sets being generated. After the data were discretised, we applied the K2 network structure learning algorithm. The K2 algorithm takes as an input a Bayesian variables ordering. For a network of size 10, this resulted in $9! = 40,320$ different orderings. We assumed that durational variable *D* is always the last in the ordering. We applied the K2 algorithm to each of the 27 discretised data sets. For the BN of size 10, we should have

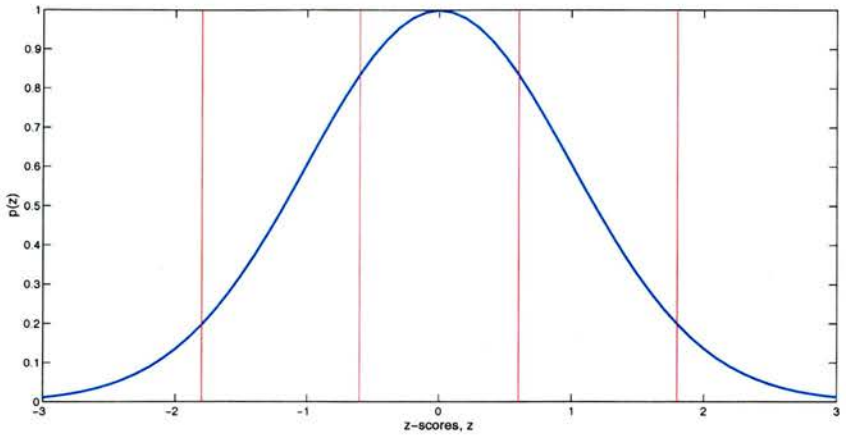

Figure 5.1: Discretisation of the normalised durations; number of bins is 5. The discretisation is performed under assumption of normalised durations following Gaussian distribution.

run the K2 algorithm $40,320 \times 27 = 1,088,640$ times, would it not have been computationally prohibitive. Hence, we decided to run it in a *orderings*-wise and *discretisation*-wise manner. We chose one variable ordering, i.e. $\pi = \{$ *Wpos, Front, Height, Wd, S, Length, Rnd, Utt, Cpos, S* $\}$, for which we ran the search procedure for 9 discretisation levels. In addition, we ran the K2 algorithm for all $40,320$ variable orderings for one discretisation level (number of bins 5). Hence, we ran it $40,320 \times 3 = 120,960$ times. We found 947 unique networks of size 10.

5.2.3 Choosing unique networks

We divided all the networks learnt by the K2 algorithm into the classes based on the parent sets of the duration node D. We call such a grouping *duration parent set equivalence.* All the networks within a class have the same duration variable D parent set $\mathbf{Pa}(D)$, being different otherwise (the parent sets of the linguistic variables within the same class may be different). We will call such a grouping *durational parent set equivalence.* If all linguistic variables are observed, all networks within a class will give the

same conditional *pdf* for D.

We should note that the networks within each parent set class may not be Markov equivalent[1], i.e. they may differ in the set of probability distributions that they represent. Since all the unique BNs found by the K2 algorithm had the same Bayesian measure (Section 4.6.1) assigned to their DAGs (meaning the networks found describe the data the best), we chose a representative network from each durational parent set equivalence class.

For the *FHLR* models we identified 7 different network topologies. We will perform all subsequent experiments with these topologies using the original, continuously-valued, duration data. An example *FHLR* model with the

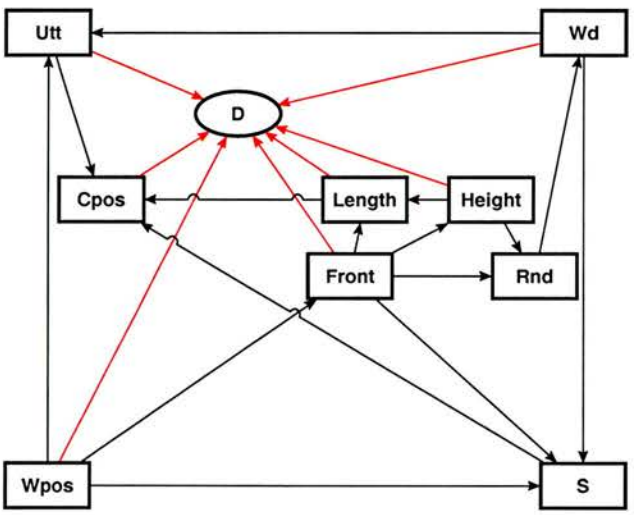

Figure 5.2: A Bayesian network of size 10 learnt by the K2 algorithm, with vowel durations being uniformly discretised. Duration D parent set $\mathbf{Pa}(D) = \{$ *Wpos, Utt, Cpos, Front, Height, Length, Wd* $\}$.

duration parent set $\mathbf{Pa}(D) = \{$ *Wpos, Utt, Cpos, Front, Height, Length, Wd* $\}$ is shown in Figure 5.2. The rest of the network examples are shown in Figures A.1-A.5 of the Appendix.

The duration variable D parent sets $\mathbf{Pa}(D)$ for the *FHLR* networks are

[1] As Chickering (2002) puts it: "Two network structures are *equivalent* if the set of distributions that can be represented by using one of the structures is identical to the set of distributions that can be represented using the other."

Name	Pa(D)	# params
VBN1-10	Cpos Length Round	80
VBN2-10	Cpos Front Length Rnd	240
VBN3-10	Cpos Front Height Length Rnd	720
VBN4-10	Cpos Front Height Length Wd	720
VBN5-10	Wpos S Cpos Rnd	120
VBN6-10	Wpos Cpos Length Rnd Wd	480
VBN7-10	Wpos Utt Cpos Front Height Length Wd	6,480

Table 5.1: Networks of size 10 learnt by the K2 algorithm, with vowel durations being uniformly discretised. The number of CG pdf parameters of the D variable is shown in the third column of the table.

shown in Table 5.1. To refer to an individual network within a group, we will use the notation given in the first column of Table 5.1: VBN*nn-NN*, where *nn* is the number assigned to the network within a group, and *NN* is the network size.

5.2.4 Model training

We performed a set of experiments, in which we trained the models and compared their performance to the baseline CART and SoP models. In addition, we wanted to find the best *FHLR* model: the one that predicts vowel duration with maximum correlation and minimum RMS error. We trained the *FHLR* networks under the *FULL* observation condition, when all discrete variables were observed, on each of the 3 voices: *lja*, *rjs*, and *erm*.

We trained each model by estimating the networks' parameters. We assumed that the values of the discrete variables (which represent linguistic factors influencing a segment's duration) follow an unrestricted multinomial distribution (discussed in Section 4.7.2). Hence, we estimated the prior values of the discrete variable's parameters as uniform Dirichlet priors with equivalent sample size of 2, according to Equation 4.27 defined on page 70. Then the maximum a *posteriori* (MAP) values of the parameters were

estimated using the EM algorithm described in Section 4.7.

It should be pointed out that we performed the network structure learning assuming the duration variable D is discrete. This was done in order to use the K2 structure learning algorithm which is only defined for discrete networks. However, from now on, we are using the original, continuously-valued consonant duration data. We assume the continuous variable for vowel duration D follows a 1-dimensional CG distribution, with probability density function defined according to Equation 4.9 on page 57.

For each instantiation $\mathbf{i} \in \mathbf{Pa}$ of the parents of duration D variable, we estimated the prior values of the duration variable D parameters $\Theta(\mathbf{i}) = (\mu(\mathbf{i}), \sigma^2(\mathbf{i}))$ as ML estimators:

$$\hat{\mu}(\mathbf{i}) = \frac{1}{K} \sum_{k=1}^{K_\mathbf{i}} \mathbf{x_k}(\mathbf{i}) \tag{5.4}$$

$$\hat{\sigma}^2(\mathbf{i}) = \frac{1}{K_\mathbf{i}} \sum_{k=1}^{K_\mathbf{i}} (\mathbf{x_k}(\mathbf{i}) - \hat{\mu}(\mathbf{i}))(\mathbf{x_k}(\mathbf{i}) - \hat{\mu}(\mathbf{i}))^T \tag{5.5}$$

where for each instantiation $\mathbf{i}$ of the discrete parents $\mathbf{Pa}(D)$ in the train set, we calculated the mean $\mu(\mathbf{i})$ and standard deviation $\sigma^2(\mathbf{i})$ of vowel duration; $\mathbf{x_k}(\mathbf{i})$ is the k-th feature vector of the training database $\mathcal{D}_M$, $K_\mathbf{i}$ is the total number of feature vectors $\mathbf{x_k}(\mathbf{i})$ in the training database $\mathcal{D}_M$.

5.2.5 *FULL* observation results

Figure 5.3 shows the correlation and RMS error results (in logarithm of ms) for the *FHLR* networks trained under the *FULL* observation condition. Table 5.2 also summarises the same results. We compared the correlation and RMS error results against two baseline models: the sums-of-products model *SoP-vowels-7* described in Section 3.5.1 on page 33, and the CART model described in Section 2.6 on page 15. We performed 2 paired t-tests comparing the correlations of each of the *FHLR* models to those of the *SoP-vowels-7* model and CART model, respectively. As it follows from the t-tests results, the VBN4-10 and VBN5-10 models significantly ($t_2 = 3.5; p < 0.05$, one-tailed) outperform the sums-of-products *SoP-vowels-7* model. All

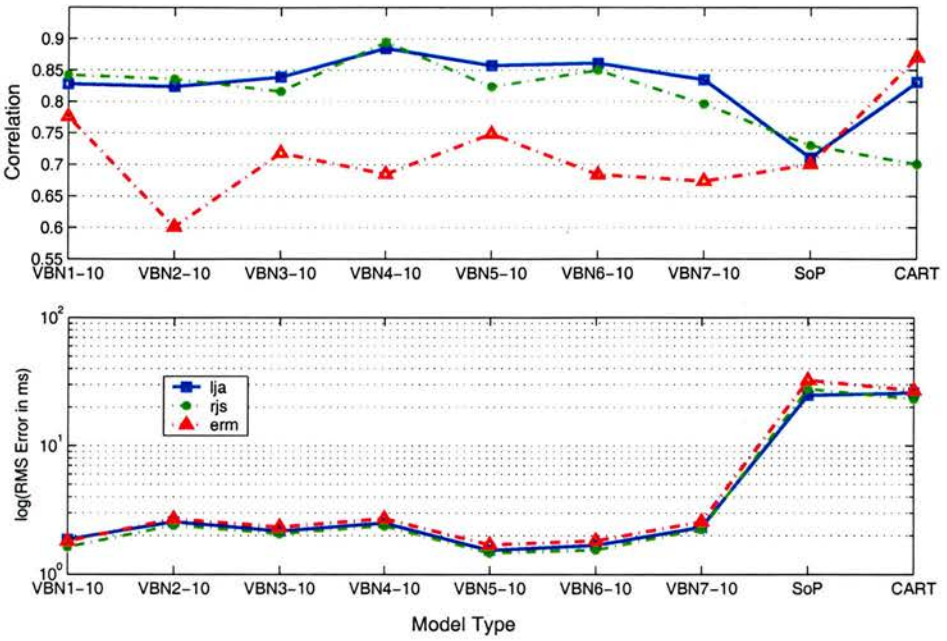

Figure 5.3: Test sample correlation and RMS error by model by voice. *FHLR* networks. *FULL* observation condition.

FHLR models perform no worse than the CART models, with the VBN3-10 model being the closest to the CART model in terms of the correlation results ($t_2 = -0.754; p > 0.001$, one-tailed). The paired t-tests comparing the performance of the *FHLR* models in terms of the RMS error revealed that all the models significantly ($p < 0.001$, one-tailed) outperform both the SoP and CART models. Given that the VBN3-10 and VBN5-10 models predict vowel duration with the maximum correlation (across all models) and quite low RMS errors, we chose the VBN3-10 and VBN5-10 models as the best *FHLR* models for the RP *lja, rjs* and GA *erm* voices, respectively.

Model	Correlation			RMSE		
	Voice					
	lja	rjs	erm	lja	rjs	erm
VBN1-10	0.834	0.796	0.673	2.30	2.22	2.55
VBN2-10	0.860	0.849	0.683	1.67	1.53	1.81
VBN3-10	**0.884**	**0.894**	0.684	2.48	2.35	2.69
VBN4-10	0.857	0.823	0.748	**1.52**	**1.45**	**1.68**
VBN5-10	0.827	0.841	**0.775**	1.85	1.61	1.79
VBN6-10	0.839	0.816	0.718	2.14	2.04	2.31
VBN7-10	0.823	0.835	0.600	2.54	2.37	2.67
SoP-vowels-7	0.71	0.72	0.70	24.5	27.8	32.1
CART	0.861	0.88	0.89	25.7	23	26.9

Table 5.2: The correlation and RMS error results by model type and voice. *FHLR* networks, *FULL* observation condition. The maximum (minimum), across different Bayesian models, correlation (RMS error) values are shown with a **boldface**.

5.3 *FH-compound* networks

5.3.1 Selecting variables for Bayesian domain set

In Section 5.2 we considered the *FHLR* models whereby the vowel identity was represented as a 4-feature entity: the features being frontness, height, length, and roundness. Given the results of the preliminary study described in Goubanova (2003) and Section 5.2, we decided to use a more compact representation for vowel identity as a 2-feature entity. First, we did without the length *Length* variable, since dropping this multi-valued variable substantially decreased the size of the feature space, and consequently, the number of network parameters that have to be estimated. Second, we decided to use the compound frontness-height *FH* variable instead of separate frontness *Front* and height *Height* variables. Hence, we represented the vowel identity with 2 variables: the compound frontness-height *FH* and roundness *Rnd* variables. The encoding of the frontness-height *FH* variable is shown in Table 3.2 on page 3.2.

Given this new vowel identity representation, we defined the 8-variable domain set:

$\mathbf{U_8}$ = { *FH, Rnd, Wpos, S, Utt, Cpos, Wd, D* }, where the discrete (linguistic) variable names and example values are shown in Table 3.1 on page 21. We will call the networks based on the domain set $\mathbf{U_8}$, the *FH-compound* networks.

5.3.2 Learning network structure

The structure learning procedure was essentially similar to the one described in Section 5.2.2 for the *FHLR* networks. We discretised the vowel durations, which resulted in 27 discretised data sets being generated. For the *FH* compound networks we generated $7! = 5,040$ different variable orderings. For each of the 27 datasets we applied the K2 structure learning algorithm for each variable orderings, thus running the procedure $27 \times 5,040 = 136,080$ times. We found 590 unique *FH-compound* networks). Table 5.3 shows the

BN size	**# of orderings**	**# unique BNs**
8	5,040	590
10	40,320	947

Table 5.3: The breakdown of the number of variables' orderings generated

number of possible orderings as well the number of unique networks found for the *FHLR* and *FH-compound* networks. Based on the *duration parent set equivalence* discussed in Section 5.2.3 on page 77, we identified 4 different network topologies. The example *FH-compound* network with the duration parent set $\mathbf{Pa}(D)$ = { *Wpos, S, Utt, Cpos, FH, Rnd, Wd* } is shown in Figure 5.4. The rest of the networks are shown in Figures B.1-B.3 of the Appendix on pages 135-136. Table 5.4 shows the duration D parent sets $\mathbf{Pa}(D)$ for each of the 4 networks learnt by the K2 algorithm. The number of parameters of the duration variable is shown in the third column.

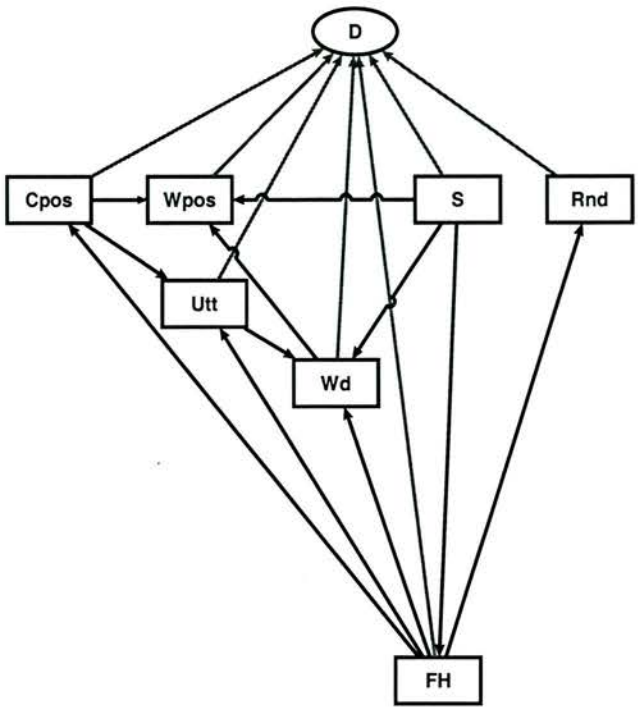

Figure 5.4: A Bayesian network learnt by the K2 algorithm, with vowel durations being uniformly discretised. The duration D parent set $\mathbf{Pa}(D) =$ { *Wpos, S, Utt, Cpos, FH, Rnd, Wd* }.

Name	Pa(D)	# params
VBN1-8	Cpos FH	90
VBN2-8	Cpos FH Rnd	180
VBN3-8	Wpos S Cpos Rnd	120
VBN4-8	Wpos S Utt Cpos FH Rnd Wd	6,480

Table 5.4: Networks of size 8 learnt by the K2 algorithm, with vowel durations being uniformly discretised. The number of CG pdf parameters of the D variable is shown in the third column of the table.

5.3.3 Model Training

We trained each of the 4 *FH-compound* networks under 2 different conditions:

1. fully observed condition (*FULL*), when all the discrete variables were

observed during training

2. partially observed condition, when some of the discrete variables were hidden; we call this the *HIDDEN* variables condition

We estimated the networks' parameters as described in Section 5.2.4 for the *FHLR* networks. After the training, we predicted the vowel durations of the test set for each of the 4 *FH-compound* networks. We compared the prediction results with those of the sums-of-products and CART model.

5.3.4 *FULL* observation results

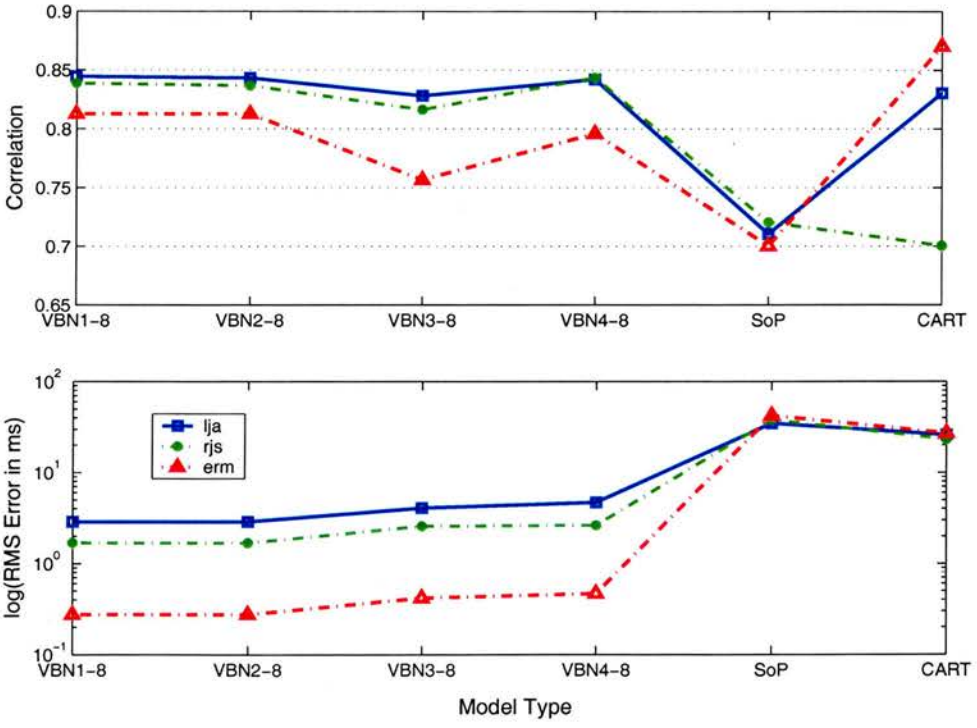

Figure 5.5: The test sample correlation and RMS error results by model type by voice. *FH-compound* networks. *FULL* observation condition.

Figure 5.5 shows the correlation and RMS error results for the *FH-compound* networks trained under the *FULL* observation condition. Table 5.5 also summarises the correlation and RMS error results for the *FH-compound* networks compared against the SoP and CART models.

Model	Voice					
	lja	rjs	erm	lja	rjs	erm
VBN1-8	**0.844**	0.838	**0.812**	2.82	1.62	**2.75**
VBN2-8	0.843	0.836	0.812	**2.80**	**1.61**	**2.75**
VBN3-8	0.828	0.816	0.756	4.0	2.51	4.17
VBN4-8	0.842	**0.843**	0.796	4.1	2.4	4.65
SoP-vowels-7	0.710	0.720	0.700	24.5	27.8	32.2
CART	0.864	0.840	0.890	25.7	23.0	26.9

Table 5.5: The correlation and RMS error results by model type and voice. *FH-compound* models. *FULL* observation condition. The maximum correlation (minimum RMS error) values are shown in **boldface**.

We performed the paired t-tests comparing the correlations (and RMS errors) of the *FH-compound* models against the baseline SoP and CART models. All *FH-compound* models proved to perform significantly ($p < 0.01$, one-tailed) better than the SoP model and no worse than the CART model. In particular, the VBN1-8 and VBN2-8 models outperform the SoP model at a higher significance level of $p < 0.001$. Hence, we selected these two models as the best *FH-compound* model candidates.

5.3.5 *HIDDEN* condition results

In order to find out how the state (observed or hidden) of the linguistic variables affects the model's training and consequently, the model's prediction power, we performed model training under the *HIDDEN* variables condition. We took a minimalist approach by selecting only single and pairs of the linguistic variables to be hidden, since performing Bayesian inference on the exhaustive subsets of all possible combinations of variables is computationally prohibitive. We selected 28 *HIDDEN* variables conditions, when some of the discrete variables were hidden with the rest being observed. The *HIDDEN* variables conditions are shown in Table 5.6.

We estimated the *FH-compound* models' parameters similar to the procedure described for the *FHLR* models (see Section 5.2.4 on page 79). After

k	Hidden	k	Hidden
1	Rnd	15	Wpos-Utt
2	Rnd-Wd	16	Utt-Rnd
3	FH	17	Utt-S
4	FH-Rnd	18	Utt
5	FH-S	19	S
6	FH-Wd	20	Wpos-Cpos
7	FH-Wpos	21	Utt-Wd
8	FH-Cpos	22	Cpos
9	Wpos-Rnd	23	Cpos-Wd
10	Utt-FH	24	Wpos-Wd
11	Utt-Cpos	25	Wpos
12	Wd	26	S-Cpos
13	S-Rnd	27	S-Wd
14	Cpos-Rnd	28	S-Wpos

Table 5.6: Hidden variables chosen for the vowel Bayesian training under hidden variables condition. The pair of hidden variables is delimited with the *dash* character -. The variable names are shown in Table 3.1.

training, we predicted duration of each vowel token of the test set, as the value with the maximum probability given the values of the discrete variables. The complete correlation and RMS error results for each of the 4 *FH-compound* networks trained under each of the 28 *HIDDEN* variables conditions are shown in Tables E.1-E.4 of the Appendix on pages 148-151.

To quantify the effect of the *HIDDEN* variables conditions upon the performance of the *FH-compound* models, we used the paired t-test, whereby for each of the 4 *FH-compound* networks we compared the correlation (RMS error) difference resulted from the training under the *FULL* and *HIDDEN* variables conditions. Our null hypothesis H_0 states that there will be no difference in the sample correlation (RMS error) between the *FULL* and and any of the k ($k = 1, \ldots, 28$) *HIDDEN* variables conditions. The non-null hypotheses H_1^k ($k = 1, \ldots, 28$) states that there will be a significant difference ($p < 0.01$, one-tailed) in the sample correlation (RMS error) between

the two experimental set-ups.

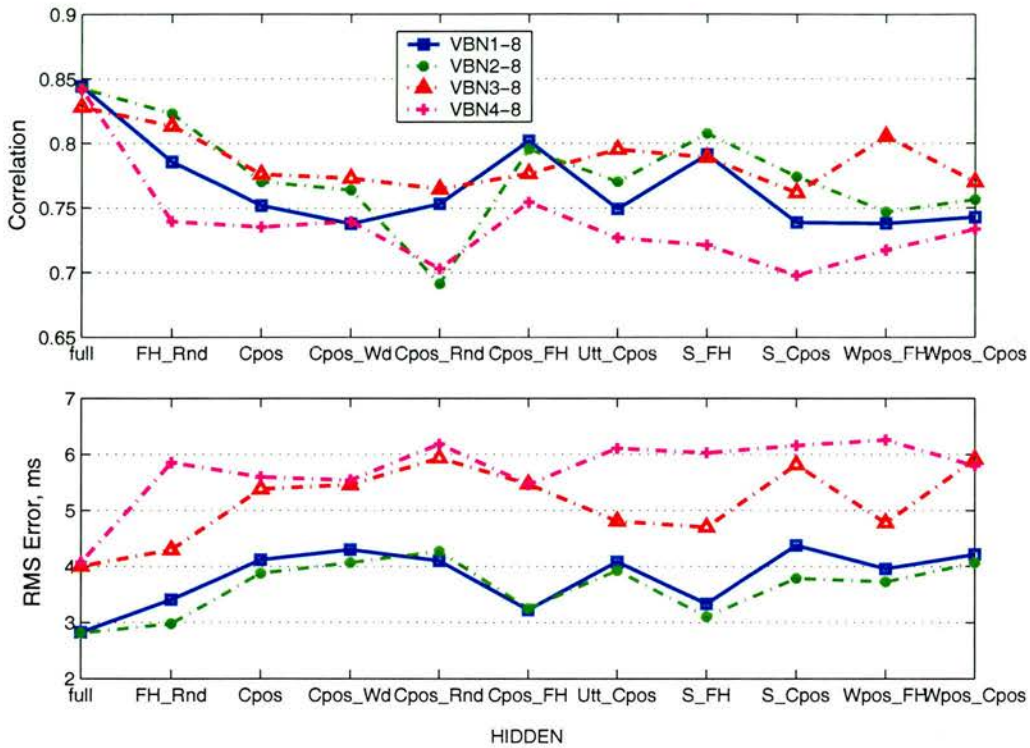

Figure 5.6: RP English; **lja** female voice; test size 3,876 vowels. Test sample correlation (RMS error) by hidden nodes. *FH-compound* networks. *HIDDEN* variables condition.

Figures 5.6-5.8 show the correlation and RMS error results for some of the significant *HIDDEN* variables conditions. As can be seen from Figures 5.6-5.8 and Tables E.1-E.4, the correlation values vary widely, depending on the *HIDDEN* variables condition. The correlation change from 0.738 to 0.844 for the VBN1-8, from 0.62 to 0.843 for the VBN2-8, from 0.622 to 0.828 for the VBN3-8, and from 0.449 to 0.843 for the VBN4-8 models, respectively. The RMS error values range from 1.6 to 8.8 ms for the VBN1-8, from 1.6 to 4.5 ms for the VBN2-8, from 2.5 to 5.9 ms for the VBN3-8, and from 2.6 to 9.2 ms for the VBN4-8 models respectively.

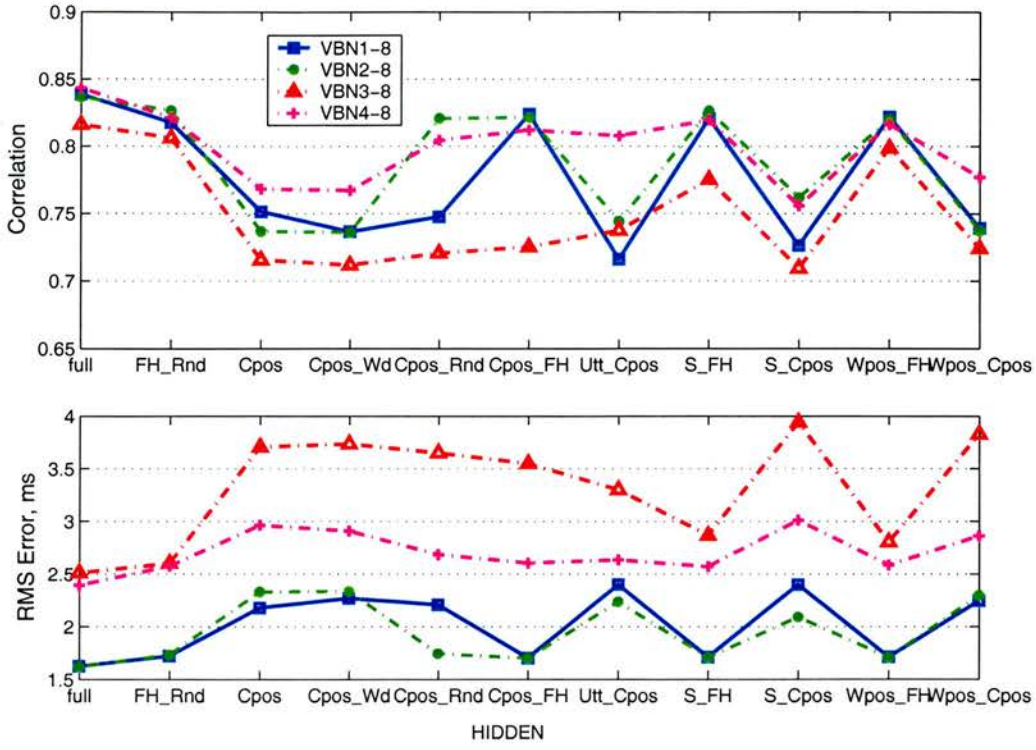

Figure 5.7: RP English; **rjs** male voice; test size 9,766 vowels. Test sample correlation (RMS error) by hidden nodes. *FH-compound* networks. *HIDDEN* variables condition.

5.3.6 Discussion

We performed a paired t-test in order to quantify the effect of the models' performance under *HIDDEN* variables condition. Tables C.1-C.4 on pages 137-140 of the Appendix show the complete t-test results for the correlation. Tables D.1-D.4 on pages 142-145 of the Appendix show the results for the RMS error. The summary of these results for the *FH-compound* models is shown in Table 5.9, where the *HIDDEN* variables conditions that result in significant ($p < 0.01$, one-tailed) changes in the correlation (RMS error) are marked with a tick $\surd$.

It should be pointed out that some of the *HIDDEN* conditions are deemed to be redundant (no changes in correlation or RMS error would

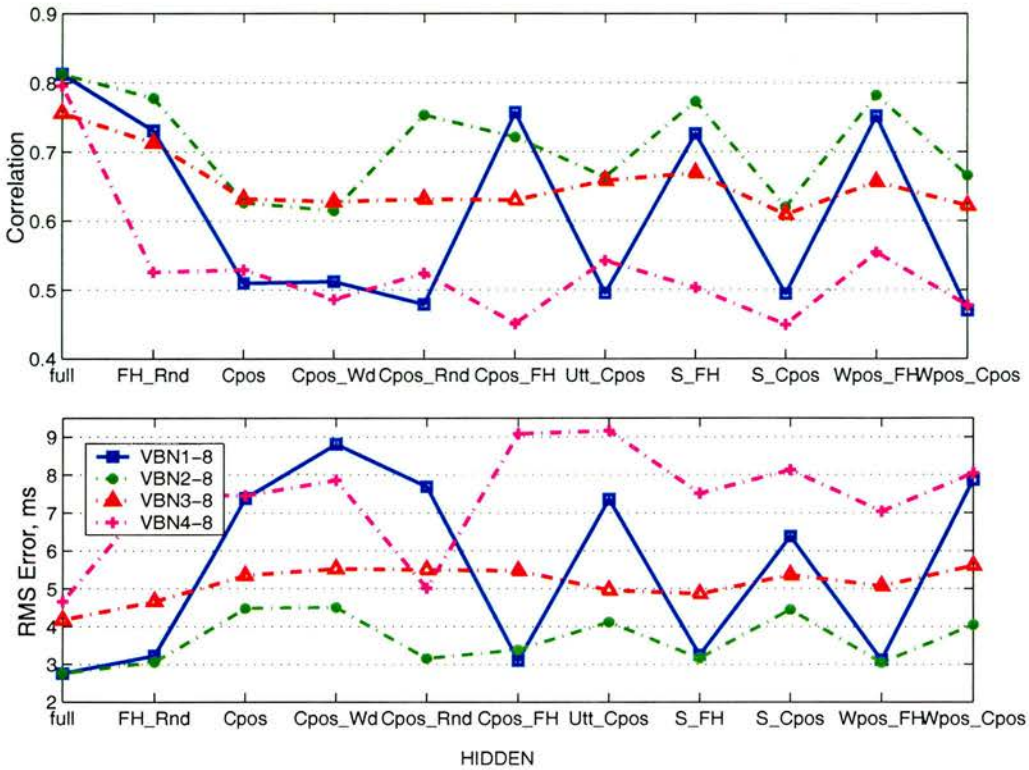

Figure 5.8: GA English; **erm** male voice; test size 6,084 vowels. Test sample correlation (RMS error) by hidden nodes. *FH-compound* networks. *HIDDEN* variables observation.

be occur) due to the structure of a particular network. For example, the VBN1-8 model has 2 duration D variable parents: the frontness-height FH and the following segment identity $Cpos$ as can be seen from Figure B.1 on page 135 and Table 5.4 on page 84. If both the frontness-height FH and the following segment identity $Cpos$ are observed, the duration D variable will be blocked from the rest of the linguistic variables, and therefore the evidence on any of these variables will not affect the duration D. Such redundant *HIDDEN* conditions, when all parents of the duration D are instantiated, are marked with a star $\star$ sign in Table 5.9. For the redundant conditions we will expect no changes in the models' performance. However, if any of the parents of the duration D variable is hidden, then we may expect the

k	Hidden	Model			
		VBN1-8	VBN2-8	VBN3-8	VBN4-8
1	Wd	⋆	⋆	⋆	⋆
2	Rnd	⋆	√		
3	Rnd Wd	⋆			
4	FH			⋆	
5	FH Wd			⋆	
6	FH Rnd				
7	Cpos			√	
8	Cpos Wd			√	
9	Cpos Rnd			√	
10	Cpos FH			√	
11	Utt	⋆		⋆	
12	Utt Wd	⋆		⋆	
13	Utt Rnd	⋆	⋆		
14	Utt FH				
15	Utt Cpos	√	√	√	
16	S	⋆	⋆		√
17	S Wd	⋆	⋆		√
18	S Rnd	⋆	⋆	√	
19	S FH	⋆			
20	S Cpos			√	
21	S Utt	⋆	⋆	√	
22	Wpos	⋆	⋆		
23	Wpos Wd	⋆	⋆		
24	Wpos Rnd	⋆			
25	Wpos FH				
26	Wpos Cpos	√	√	√	
27	Wpos Utt	⋆	⋆		
28	Wpos S	⋆	⋆		√

Table 5.7: The paired (*FULL* vs. *HIDDEN*) t-test results for the correlation (RMS error). *FH-compound* networks. The t-test significant ($p < 0.01$) pairs are marked with a tick √. The t-test redundant pairs are shown with a star ⋆. The pairs' names are shown in Table 5.6.

correlation and RMS error will change, given a particular model's structure.

For the VBN1-8 model, for example, whenever either the frontness-height *FH* or the following segment identity *Cpos* are hidden, we will expect the correlation and RMS error changes for the corresponding *HIDDEN* conditions.

As it turns out, there are no significant ($p < 0.01$, one-tailed) changes in the correlation values when the VBN1-8 model was trained under either of the 28 *HIDDEN* variables conditions. However, there are two *HIDDEN* variables conditions that resulted in increased RMS error. When the following segment identity *Cpos* and either the within-word position *Wpos* or the within-utterance position *Utt* variables are hidden during training, the VBN1-8 model predicts vowel duration with significantly higher RMS error values than those for the *FULL* observation condition. Overall, we may conclude that the VBN1-8 model trained under any of the 28 *HIDDEN* variables conditions predicts vowel duration without any significant decrease in the correlation. However, the model predicts the test set vowel durations with a significantly higher RMS error, if a vowel's position within the word or utterance, as well as its following segment identity are not known.

Compared to the VBN1-8 model, the duration *D* node of the VBN2-8 model has one more parent: the roundness *Rnd* node (see Figure B.2 on page 136 of the Appendix). When the following segment identity *Cpos* and the within-word position *Wpos* are hidden, the VBN2-8 predicts vowel duration with significantly lower correlation values than those for the *FULL* variables condition. When the roundness *Rnd* variable is hidden, the test set vowel durations are predicted with significantly higher RMS error values than those for the *FULL* observation condition.

Overall, the VBN2-8 model performs significantly worse when when the following segment identity and either the within-word or within-utterance position are hidden. In addition, when the vowel's roundness is is not known, the model's performance significantly degrades

As can be seen from Table 5.4, the duration *D* variable parent set of the

VBN3-8 model consists of 4 parents: the within-word position *Wpos*, the stress *S* the roundness *Rnd* and the following segment identity *Cpos* variables. This is a more connected model, and it turns out that when the following segment identity *Cpos* and any of the rest of the linguistic variables are hidden, the VBN3-8 model predicts vowel duration with significantly higher RMS error values compared to one for the *FULL* observation condition. The VBN3-8 model trained with the roundness *Rnd* and either the stress *S* or the following segment identity *Cpos* being hidden predicts vowel duration with significantly lower correlation values compared to those for the *FULL* observation condition. Overall, the VBN3-8 model is more sensitive to the state of the following segment identity *Cpos* combined with the rest of the duration variable parents. In addition, when the stress and the within-utterance position are hidden, the model's performance significantly degrades too.

The VBN4-8 model has all linguistic variables connected to the duration *D* node. The model turns out to be very sensitive to the state of the stress *S* variable. When the state of the stress *S* is hidden, or both the stress *S* and either the within-word position *Wpos* or the word class *Wd* variables are hidden, the VBN4-8 model predicts vowel duration with significantly lower correlation values than those for the *FULL* observation condition. However, in terms of the RMS error, the model's performance is no worse compared to the *FULL* observation condition.

5.4 Choosing the best model

Based on the results of the training under the *FULL* observation condition we selected our best *FHLR* model. For the RP *lja, rjs* voices we chose the VBN3-10 and VBN5-10 models as the best models. In terms of the correlation, they all perform better than the SoP model and no worse than the CART model. For example, the VBN3-10 model predicts vowel duration with the correlation of 0.884 vs. 0.823-0.835 for the SoP and 0.861-0.88 for

the CART models respectively. In terms of the RMS error, the VBN3-10 model significantly ($p < 0.01$, one-tailed) outperforms both the SoP and CART models. For the GA *erm* voice we selected the VB5-10 model. It performs with the maximum correlation of 0.775 and minimum RMS error of 1.68 ms. The VBN5-10 model outperforms both the SoP and CART models, predicting vowel duration with the RMS error of 1.68 that compares favourably against the values of 32.1 ms and 26.9 ms for the SoP and CART models respectively.

We also selected the VBN2-8 model as our best *FH-compound* model. When trained under the *FULL* observation condition, the VBN2-8 model predicts vowel duration with the correlation varying from 0.812 to 0.844 for the RP and GA voices. The model beats both SoP and CART models in terms of the RMS error. For example, for the *rjs* voice, the model's RMS error is 1.68 vs. 27.8 ms and 23 ms for the respective SoP and CART models. The VBN2-8 model demonstrates robust behaviour when trained under the 6 *HIDDEN* variables conditions. Its performance significantly degrades only when the following segment identity and either the word-level or the utterance-level information is not known. However, the model's performance does not significantly change, when the information about the following segment identity alone is hidden. In addition, when the following segment identity as well as either word class, roundness, or front-height variables are hidden, there is no loss in the model's performance.

In real TTS system, it could be potentially useful in predicting duration of a vowel followed by an ambisyllabic consonant, or in predicting duration of a syllabic consonant, since neither of the features: frontness, height, or roundness would be known. For example, in words such as *prison, bacon* or phrases such as "Jack and Jill", the vowel preceding a syllabic consonant is substituted for the syllabic consonant to become a nucleus. Therefore, all the information about the following segment identity or vowel's features (frontness, height, roundness) becomes irrelevant.

5.5 Summary

In this chapter, we discussed Bayesian models for predicting vowel duration. In particular, we considered 2 classes of models that resulted from two different representations of the vowel identity. First, we represented the vowel identity as a 4-feature entity based on frontness, height, length, and roundness phonological distinctive features. Second, we represented the vowel identity as a 2-feature entity. We discarded the length *Length* variable, and combined frontness *Front* and height *Height* variables into one compound front-height *FH* variable. This way we reduced the number of the candidate network structures and the number of the Bayesian parameters to be learnt.

Given the 4-feature vowel identity representation, we defined a 10-variable Bayesian domain $\mathbf{U}_{10}$ discussed in Section 5.2.1. Given the 2-feature vowel identity representation, we defined a 8-variable Bayesian domain $\mathbf{U}_8$ discussed in Section 5.3.1. We called the class of models for the $\mathbf{U}_{10}$ domain the *FHLR* models; and the class of models for the the $\mathbf{U}_8$ domain the *FH-compound* models.

We applied the K2 structure learning algorithm to the data described according to these two domains. For the $\mathbf{U}_{10}$ domain we identified 7 *FHLR* unique networks, and for the $\mathbf{U}_8$ domain 4 *FH-compound* unique networks.

Following the structure learning, we performed model training. We calculated duration variable parameters as ML estimators of the one-dimensional GC distribution. We applied the EM algorithm to calculate the parameters of the discrete (linguistic) variables as the MAP estimates of the multinomial distribution with the Dirichlet priors. For the *FHLR* models we learnt the models' parameters under the *FULL* observation condition. For the *FH-compound* models we learnt the the models' parameters under *FULL* and *HIDDEN* variables conditions. After training we performed inference on the test set. For each vowel token of the test set we predicted its duration as the value with the maximum probability given the values of the discrete variables.

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
VBN3-10	**0.884**	**0.894**	0.684	2.48	2.35	2.69
VBN5-10	0.827	0.841	**0.775**	1.85	1.61	1.79
VBN2-8	0.843	0.836	0.812	**2.81**	**1.61**	**2.75**
SoP-vowels-7	0.71	0.72	0.70	24.5	27.8	32.1
CART	0.861	0.88	0.89	25.7	23	26.9

Table 5.8: The correlation and RMS error results by voice. The best *FHLR* and *FH-compound* networks. The *FULL* observation condition.

Table 5.8 shows the duration prediction results for the best *FHLR* and *FH-compound* models trained under the *FULL* observation condition. Based on the paired t-test results, the VBN3-10 and VBN5-10 were were found to perform significantly ($p < 0.01$) better than the SoP and no worse than the CART model. Likewise, the VBN1-8 and VBN2-8 models were found to outperform the SoP model, and perform no worse than the CART model.

We also trained the *FH-compound* models under various *HIDDEN* variables conditions. The reason for doing this was to see how the state (hidden/observed) of the linguistic variables influences the performance of the models. We analysed the effect of the *HIDDEN* conditions on the models' performance in terms of the test sample correlation and RMS error for each of the 4 *FH-compound* models.

Table 5.9 summarises the t-test results for the correlation and RMS error. The *HIDDEN* variables conditions that result in significant ($p < 0.01$; one-tailed) decrease in correlation and increase in RMS error are marked with a tick $\surd$. As can be seen from the summary table, the VBN1-8 model's performance degrades whenever the following segment identity and either the word or utterance-level positional variables are hidden. The VBN2-8 model predicts vowel duration with lower correlation (higher RMS error) when the roundness is hidden, or when the following segment identity and either the word or utterance-level positional variables are hidden. The performance of the VBN3-8 model degrades significantly whenever the following segment

k	Hidden	Model			
		VBN1-8	VBN2-8	VBN3-8	VBN4-8
2	Rnd	⋆	√		
7	Cpos			√	
8	Cpos Wd			√	
9	Cpos Rnd			√	
10	Cpos FH			√	
15	Utt Cpos	√	√	√	
16	S	⋆	⋆		√
17	S Wd	⋆	⋆		√
18	S Rnd	⋆	⋆	√	
20	S Cpos			√	
21	S Utt	⋆	⋆	√	
26	Wpos Cpos	√	√	√	
27	Wpos Utt	⋆	⋆		
28	Wpos S	⋆	⋆		√

Table 5.9: The summary of the paired (*FULL* vs. *HIDDEN*) t-test results for the correlation (RMS error). The *HIDDEN* variables conditions that resulted in significant ($p < 0.01$, one-tailed) decrease in the correlation (increase in the RMS error) are marked with a tick √. The redundant *HIDDEN* variables conditions are marked with a star ⋆. The *FH-compound* networks. The pairs' names are shown in Table 5.6.

identity and either of the rest of the parents of the duration variable are hidden. Additionally, the model is sensitive the state of the stress S variable. The VBN4-8 model predicts vowel duration with lower correlation (higher RMS error) when the stress S variable and either the within-word position *Wpos* or the word class *Wd* are hidden.

Based on the results of this chapter, we selected the VBN3-10 and VBN5-10 to be the best *FHLR* models for the RP *lja, rjs* voices and the GA *erm* voices respectively, since these models predict vowel duration with the maximum correlation and minimum RMS error. They also outperform the SoP model and performs no worse than the CART model.

We selected the VBN2-8 model as the best *FH-compound* model. The VBN2-8 model requires quite small number (180) of the duration D variable parameters to be estimated. The model performs significantly better than the SoP model and no worse than the CART model when trained under *FULL* observation condition. The VBN2-8 model is quite robust in terms of the correlation and RMS error changes when trained under the *HIDDEN* variables condition.

The VBN2-8 model performance does not degrade when the information about the following segment identity variable is not known. This property of the model may be used in the real TTS system, when predicting duration of a vowel followed by an ambisyllabic consonant. When the following segment identity as well as either word class, roundness, or front-height variables are hidden, there is no loss in the model's performance. This may be also beneficial when predicting duration of a syllabic consonant, since neither front-height, nor roundness features information will be relevant.

In general, the *FHLR* and *FH-compound* Bayesian models for vowels, with the former based on a 4-feature and the latter based on a 2-feature vowel identity representation, outperform both the sums-of-products and CART models in terms of the RMS error. In terms of the correlation, our best models, i.e the VBN3-10, VBN5-10, and VBN2-8 perform better than the sums-of-products model and no worse than the CART model. They can be successfully implemented in any real TTS system.

Chapter 6

Bayesian Models: Consonants

In this chapter we present a Bayesian model for predicting consonant duration. We will use a method similar to the one for predicting vowel duration described in Section 5.1. We start this chapter with presenting our preliminary experiments and results of consonant duration prediction using a simple belief network described in Section 6.1. We discuss the issues involved in improving this simple model by introducing other linguistic factors that are known to influence consonant duration in Section 6.2. Thus we define a new class of models that we call *MV-compound* models: in this class of models the consonant identity is represented with the manner of production and voice distinctive features. We discuss the structure learning algorithm applied to consonant data in Section 6.3. We proceed with describing the model training in Section 6.4. Following this, the results of the experiments are presented and discussed in Section 6.5. In evaluating the performance of the Bayesian networks for consonants we use the same metrics as for vowels: sample correlation and RMS error described in Chapter 5, Sections 5.1.1-5.1.2. We conclude this chapter with a summary of the results in Section 6.7.

6.1 Preliminaries

Variable	# Values	Example
consonant identity C	24	/ch/
within-word position *Wpos*	3	initial
stress S	2	stressed
frontness *Front*	3	back
number of the syllables in word *NSyls*	5	2

Table 6.1: Linguistic factors chosen for the CBN1-6 model.

Before defining a fully-fledged Bayesian model for consonants, we decided to perform a pilot study with a simpler model. This study was part of the work presented in Goubanova (2001). We will present this work in more detail in this section. Based on the research by van Son & van Santen (1997) we selected 5 linguistic factors that are known to influence a consonant's duration. These were the consonant identity C, the within-word position *Wpos*, the stress level of the syllabic vowel S, the number of the syllables in the word *NSyls*, and the frontness of the syllabic vowel *Front*. The variable names, their encoding, and example values are shown in Table 6.1.

Given these factors, we defined a belief network that consisted of 6 variables: 5 discrete variables for the linguistic factors and one continuous variable for the consonant duration. Hence, the network has the following domain set: $\mathbf{U_6} = \{C, Wpos, S, NSyls, Front, D\}$. The graphical representation of this model that we call the CBN1-6 model, is shown in Figure 6.1. We did not perform the network structure search in this experiment. We devised a network structure by hand, so that it encoded some of the linguistic variables' interactions. For example, as can be seen from the figure, the interaction between the within-word position *Wpos* and the syllabic stress S is represented in the DAG structure with an arc going from *Wpos* to S: $Wpos \rightarrow S$. We also assumed that all the linguistic variables are directly connected to the durational variable D. Figure 6.2 and Table 6.2 show the

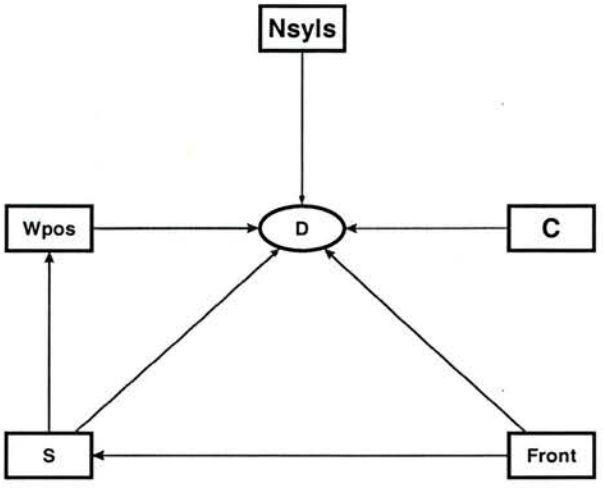

Figure 6.1: Bayesian network of size 6 for consonant duration prediction; the parent set $\mathbf{Pa}(D) = \{C, Wpos, S, NSyls, Front\}$.

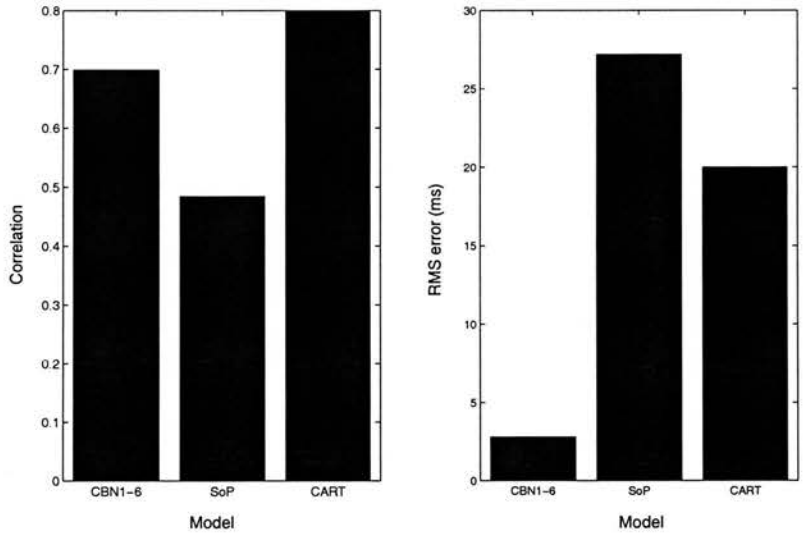

Figure 6.2: Test set RMS error (ms) results by model type. The **rjs** RP male voice (financial database). The test set size: 7,110 consonants.

test set correlation and RMS error results for the *CBN1-6*, SoP, and CART models, respectively. In terms of the correlation, the Bayesian model performs better than the SoP model, but significantly ($p < 0.05$, one-tailed)

Model	Correlation	RMS error
CBN1-6	0.699	2.8
SoP-vowels-6	0.484	27.2
CART	0.80	20

Table 6.2: The correlation and RMS error results by model type. The **rjs** RP male voice (financial database). The test set size: 7,110 consonants.

worse than the CART model. In terms of the RMS error, our Bayesian model performs significantly ($p < 0.05$, one-tailed) better than both the SoP and CART models.

Given such a poor (in terms of the correlation values) performance of the CBN1-6 model compared to the CART model, we concluded that a better and more robust Bayesian model for predicting consonant duration is needed. First, we should consider various additional factors that are known to influence consonant duration. Second, we should perform a network structure search to find the model that would the best fit to the data. We will search for a better model hoping, that it will beat the CART model.

6.2 Selecting variables for a new model

In Section 6.1 we discussed the CBN1-6 model based on 5 linguistic variables, with the consonant identity represented by the consonant type variable C. Given this multi-valued variable, the number of network parameters even for this 6-variable network that need to be estimated was quite high. For the RP voice for example, the number of the duration D node parameters was to 2,160. If we wanted to add just one more variable to our model, for example the 3-valued within-utterance position *Utt* variable, the number of the duration D node parameters would triple coming to 6,480. Hence, we decided to represent the consonant identity as manner of production and voice distinctive features using the 9-valued compound manner-voice variable MV (see Table 3.5 of Section 3.2.2). First, it seems more natural to use a variable that has an underlying linguistic interpretation (compared

to just purely enumerative consonant type variable C); this choice of the consonant identity representation is also supported by the results of van Santen (1994) on the effect of manner of production and voice on consonant duration. Second, by using a lower cardinality valued variable (9 vs. 24) we could cut down on the number of the model's parameters.

We also introduced the within-syllable position *Syl* variable that describes the position of the consonant within a syllable (*coda, onset*, or *syllabic*). In addition, we used the within-utterance position *Utt*, and the identity of the previous (following) segment *Cpre* (*Cpos*) variables, taking into account the effect of these factors on consonant duration, discussed in the literature review of Section 3.2.1.

Given the manner-voice based consonant identity representation, we defined a 9-variable domain set for our model: $\mathbf{U_9}$ = { *MV, Wpos, S, Utt, Syl, Cpre, Cpos, Front, D* }, where the discrete (linguistic) variable names and example values are shown in Table 3.1, Section 3.1. We will call the networks based on the domain set $\mathbf{U_9}$ the *MV-compound* models.

6.3 Learning Bayesian network structure

6.3.1 Applying the K2 structure learning algorithm

We discretised consonant durations similar to the procedure described in Section 5.2.2. For the *MV-compound* models we generated an exhaustive set of the variable orderings, with the total number of orderings being $8! = 40,320$ (as for vowels, we assumed that durational variable D is always the last in the ordering). Due to time limitations, we decided to run the learning procedure for one discretisation level only. We chose the level of discretisation equal to 5 (number of bins 5), since this number represents the median value for the total number of different quantisation levels that we considered (i.e. 9). This level of discretisation also provides enough probability mass in each bin, for the structure learning algorithm to work. Hence, we applied the K2 algorithm $40,320 \times 3 = 120,960$ times to our

discretised training set for each of the 3 voices (*lja*, *rjs*, *erm*). We found 915 unique networks of size 9.

6.3.2 Choosing unique networks

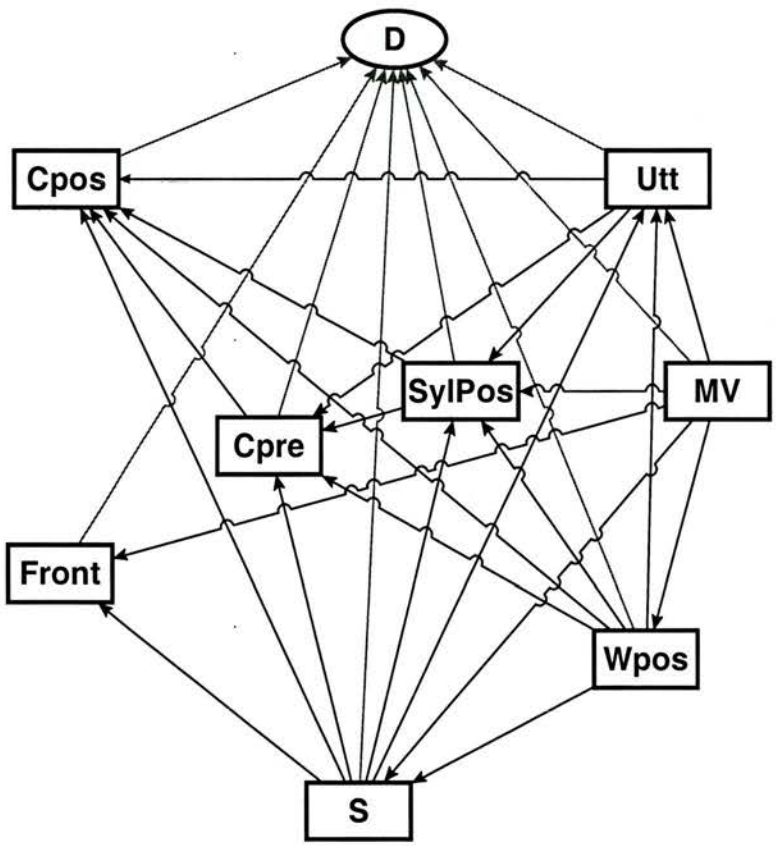

Figure 6.3: Bayesian network learnt by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* networks: CBN5 model.

Based on the *duration parent set equivalence* property introduced in Section 5.2.2, we identified 8 *parent set equivalent* classes of networks. The example network is shown in Figure 6.3. The rest are shown in Figures F.1-F.7 of the Appendix. Table 6.3 also shows the duration parent set $\mathbf{Pa}(D)$ and the corresponding number of the parameters to be estimated. To refer to the individual network within a group, we will use the notation given in the

Name	Pa(D)	# params
CBN1	*MV, Cpos*	27
CBN2	*MV, Syl, Front*	81
CBN3	*MV, Wpos, S, Syl, Cpre, Cpos, Front*	4,374
CBN4	*MV, Wpos, S, Utt, Syl, Cpre, Cpos*	4,374
CBN5	*MV, Wpos, S, Utt, Syl, Cpre, Cpos, Front*	13,122
CBN6	*MV, Wpos, Syl, Cpre, Cpos*	729
CBN7	*MV, Wpos, Syl, Cpre, Cpos, Front*	2,187
CBN8	*MV, Wpos, Utt, Syl, Cpre, Cpos, Front*	6561

Table 6.3: BNs learnt by the K2 algorithm, with consonant durations being uniformly discretised. The number of the CG pdf parameters of the D variable is shown in the third column of the table.

first column of Table 6.3, CBNnn, where nn is a number assigned to the network within a group. As for vowels, we will perform further experiments with these topologies using the original, continuously-valued, duration data.

6.4 Model training

The goal of the training experiments was to study the performance of the *MV-compound* models and to learn how the state (hidden or observed) of the discrete (linguistic) variables affects the consonant duration prediction. We trained our models under 2 different conditions: the *FULL* observation and the *HIDDEN* variables conditions.

From the literature review presented in Section 3.2.1, it follows that different linguistic factors influence consonant duration to varying degrees. For our *HIDDEN* variables condition we selected 6 partially observed conditions when a single or a pair of the discrete variables were hidden, with the rest being observed. The *HIDDEN* variables conditions are shown in Table 6.4. As can be seen from the table, we selected the within-word position *Wpos*, the syllabic position *Syl*, the previous *Cpre* and following *Cpos* segment identity variables to be hidden. We also included 2 *HIDDEN* variables conditions consisting of a pair of discrete (linguistic) variables. One was the pair of

k	Training condition	Variable(s) Hidden
0	fully observed	*full*
1	within-word position hidden	*Wpos*
2	syllabic position hidden	*Syl*
3	previous segment identity hidden	*Cpre*
4	following segment identity hidden	*Cpos*
5	syllabic position and previous segment hidden	*Syl - Cpre*
6	syllabic position and following segment hidden	*Syl - Cpos*

Table 6.4: Training conditions chosen for predicting consonant duration. The pair of hidden variables is delimited with the *dash* character -. The variables' names are shown in Table 3.4 on page 25.

the syllabic position *Syl* and previous segment identity *Cpre* variables. The other was syllabic position *Syl* and following segment identity *Cpos*. The choice of the hidden variables was dictated by the effect of the segment, syllable, and word level contextual factors on consonant duration, as follows from the results by (van Santen 1994), (Fougeron & Keating 1997), and (Turk & Shattuck-Hufnagel 2000), among others.

To check the difference in the performance between the models trained under the *FULL* and *HIDDEN* variables conditions we used a paired t-test, with the pairs being the models trained under different experimental set-ups. Our null hypothesis H_0 states that there will be no difference in the sample correlation (RMS error) between the *FULL* and any of the k ($k = 1, \cdots, 6$) *HIDDEN* variables conditions. The non-null hypotheses H_1^k ($k = 1, \cdots, 6$) state that there will be a significant difference ($p < 0.01$, one-tailed) in the sample correlation (RMS error) between the fully observed variables condition and the k-th hidden variables condition.

We trained all 8 models on each of the three voices: *lja, rjs, erm.* After the training, we performed inference on the test data. We compared the results with those of the sums-of-products and CART models.

6.4.1 Estimating the models' parameters

We trained the *MV-compound* models by estimating the models' parameters. We estimated the models' parameters as described in Section 5.2.4 for the vowel *FHLR* networks. We assumed the discrete (linguistic) variables follow a multinomial distribution, discussed in Section 4.7.2. The prior values of the parameters for the discrete (linguistic) variables were estimated as Dirichlet priors with the equivalent sample size of 2 (see Equation 4.26 on page 70). We calculated the MAP estimates of the parameters using the EM described in Section 4.7.

It should be pointed out that we performed the network structure learning assuming the duration variable D is discrete. This was done in order to use the K2 structure learning algorithm defined for discrete networks. However, we estimated the parameters of the durational variable D as a continuous variable, switching back to our initial assumptions (Section 4.4.1) about a belief network structure for predicting phone's duration. As for vowels, we assumed the consonant duration D variable follows a 1-dimensional CG distribution with probability density function defined according to Equation 4.9 defined on page 57. We estimated the prior values of the duration variable D parameters using ML estimators defined in Equation 4.21 on page 68.

6.5 Results and Discussion

6.5.1 *FULL* observation condition

Table 6.5 shows the correlation and RMS error results for the 8 *MV-compound* models trained under the *FULL* observation condition. These results are compared to those of the SoP and CART models. As can be seen from the table, across the models the correlation varies from 0.56 to 0.84, from 0.49 to 0.80, and from 0.69 to 0.80 for the *lja, rjs*, and *erm* voices respectively. The RMS error varies from 3.5 ms to 4.7 ms, from 4.1 ms to 5.6 ms, from

Model	Voice					
	lja	rjs	erm	lja	rjs	erm
CBN1	0.80	0.77	0.69	3.8	4.4	3.8
CBN2	0.73	0.76	0.67	5.1	5.6	5.1
CBN3	**0.84**	**0.80**	0.69	**3.5**	**4.1**	**3.6**
CBN4	0.72	0.74	**0.80**	4.6	5.1	4.5
CBN5	0.71	0.73	0.74	3.7	4.3	4.5
CBN6	0.80	0.74	0.75	4.6	5.2	4.6
CBN7	0.76	0.73	0.73	4.7	5.3	4.7
CBN8	0.56	0.49	0.75	3.5	4.1	3.7
SoP	0.73	0.79	0.76	25	26	33
CART	0.78	0.80	0.82	21	20	24

Table 6.5: The correlation and RMS error results by model type by voice. *FULL* observation condition.

3.6 ms to 5.1 ms for the *lja, rjs*, and *erm* voices respectively.

For the RP voices (*lja, rjs*), it is the CBN3 model (the duration parent set $\mathbf{Pa}(D)$ contains all variables except the within-utterance position *Utt*) that predicts consonant duration with the maximum correlation, beating both the SoP and CART models. For the GA *erm* voice, the CBN4 model with the duration parent set $\mathbf{Pa}(D)$ containing all variables except the frontness *Front* of the syllabic vowel, predicts consonant duration with the maximum correlation value. The CBN4 model performs better than the SoP and no worse than the CART model. In terms of the correlation, all *MV-compound* models (except for the CBN8 model) perform better than the SoP models, and no worse than the CART model. In terms of the RMS error, the *MV-compound* models beat both the SoP and CART models.

6.5.2 *HIDDEN* variables condition

Figures 6.4-6.6 show the test sample correlation and RMS error results by the *HIDDEN* variables conditions for 3 voices. The complete results are shown in Tables G.1-G.2 of the Appendix. Overall, the correlation values vary widely: from 0.49 to 0.84, from 0.40 to 0.80, and 0.37 to 0.80 for the

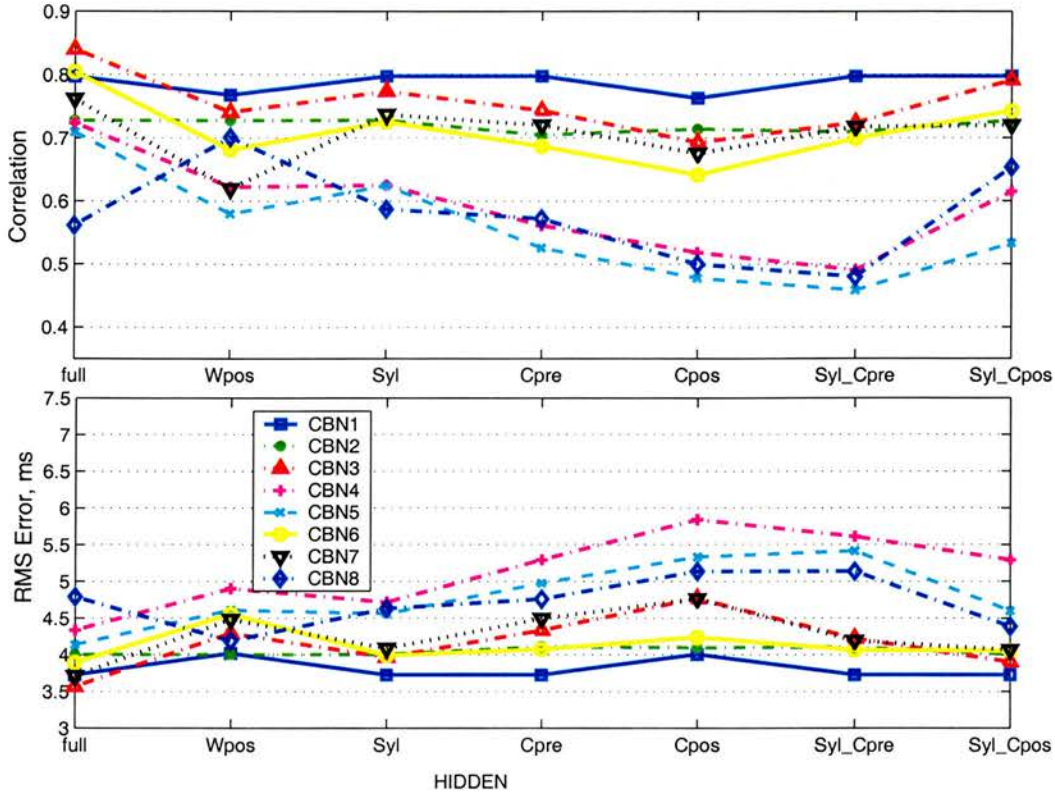

Figure 6.4: RP English; **lja** female voice; test size 6,015 consonants. Test sample correlation and RMS error by *HIDDEN* variables condition. The *MV-compound* networks for consonants.

lja, *rjs*, and *erm* voices, respectively. For the RP English *lja* female voice, the RMS error values range from 3.7 to 5.8 ms. For the RP English *rjs* male voice, the overall RMS error ranges from 3.4 to 5.5 ms. For the GA English *erm* male voice the overall the RMS error ranges from 4.1 to 8.8 ms.

6.5.3 Expected models' behaviour

We will not discuss the trivial case, whereby all the parents of the duration D variable are observed: in such cases there should be no changes in the correlation (RMS error) values. For example, in the CBN1 model (Figure B.1 on page 135) the duration D parents set consist of two variables: the manner-voice and following segment identity variables. If we trained the CBN1 model with both the manner-voice and the following segment identity variables being observed, we would expect no changes in the model's per-

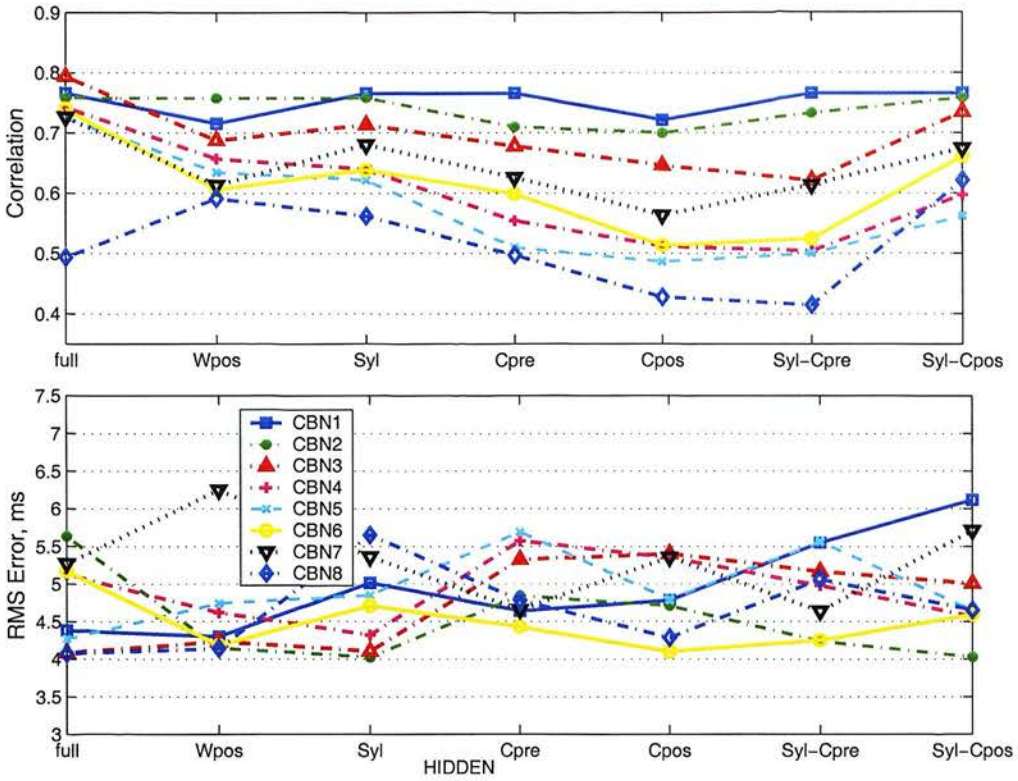

Figure 6.5: RP English; **rjs** male voice; test size 14,998 consonants. Test sample correlation and RMS error by *HIDDEN* variables condition. The *MV-compound* networks for consonants.

formance. However, if the following segment identity or the manner-voice is hidden, there will be changes in the model's performance. In any *HIDDEN* variables condition that has the following segment identity *Cpos* variable hidden, the evidence from any variable that is a parent of it (i.e. Utt, Syl, or Cpre) would affect the model's performance. In such cases, we expect changes in the model's performance.

The duration D variable of the CBN2 model has 3 parents: the manner-voice, the syllabic position *Syl*, and the frontness *Front* variables (Figure B.2 on page 136). Therefore, when the syllabic position *Syl* is hidden, or the previous *Cpre* (following *Cpos*) segment identity and the syllabic position *Syl* are hidden, we expect the CBN2 model would perform worse than the *FULL* observation condition. Compared to the CBN1 and CBN2 models, the rest of the *MV-compound* models are highly connected. As a consequence,training

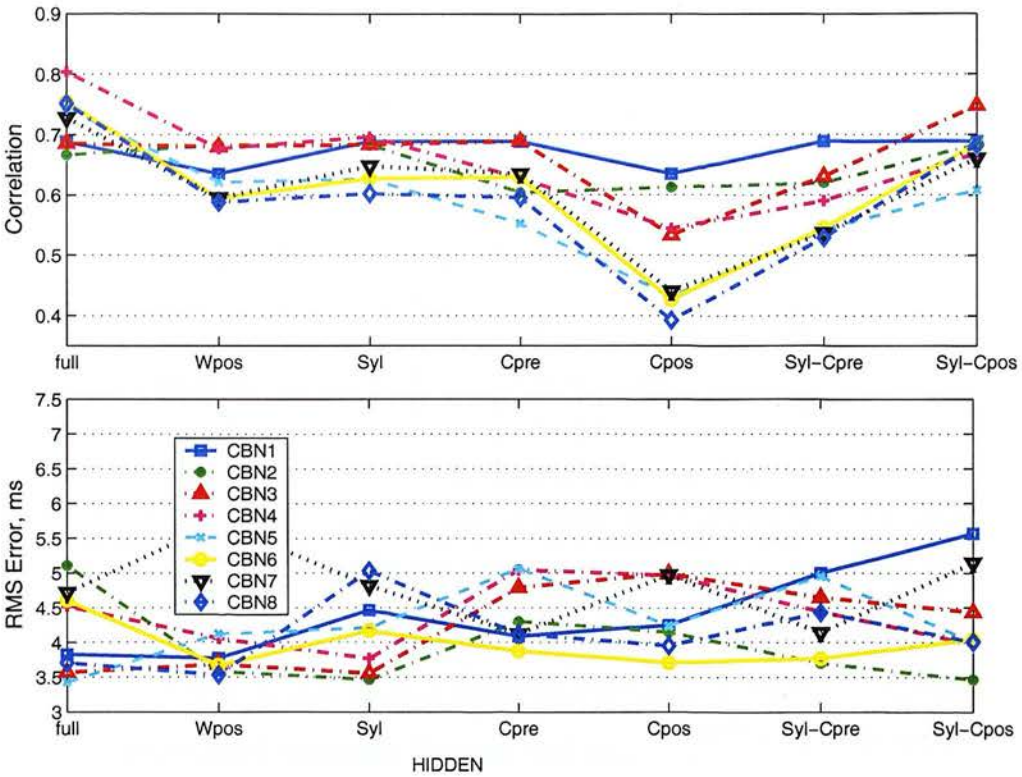

Figure 6.6: GA English; **erm** male voice; test size 9,039 consonants. Test sample correlation and RMS error by *HIDDEN* variables condition. The *MV-compound* networks for consonants.

under any of the *HIDDEN* variables conditions is expected to cause the loss in the models' performance.

6.5.4 Models' behaviour

To quantify the difference in the performance of the *MV-compound* models trained under the *FULL* and *HIDDEN* variables conditions we performed a paired t-test, whereby we compared the difference in the correlation (RMS error) between the two conditions. The complete t-test results for the correlation and RMS error are shown in Tables H.1-H.8 and Tables I.1-I.8 on pages 165-169 and 171-175 of the Appendix. Table 6.6 shows a summary of the t-test results, with the *HIDDEN* observation conditions that result in a significant ($p < 0.01$; one-tailed) decrease in the correlation (and increase in the RMS error) being marked with a tick $\surd$.

Model	HIDDEN					
	Wpos	Syl	Cpre	Cpos	Syl-Cpre	Syl-Cpos
CBN1				√		
CBN2						
CBN3				√		
CBN4	√	√	√	√	√	√
CBN5	√	√	√	√	√	√
CBN6	√					
CBN7	√		√			√
CBN8						

Table 6.6: The paired t-test results for the correlation (RMS error) values. The *HIDDEN* variables conditions that resulted in significant ($p < 0.01$, one-tailed) decrease in the correlation (increase in the RMS error) are marked with a tick √. The pairs' names are shown in Table 6.4 on page 106. *MV-compound* models.

As was expected, when the following segment identity *Cpos* was hidden, the CBN1 model predicted consonant duration with significantly lower correlation (higher RMS error) values than those for the *FULL* observation condition.

Contrary to our expectations, the CBN2 model did not perform significantly worse, when the syllabic position *Syl* variable was hidden; there were also no significant changes in the correlation (RMS error) when both the previous *Cpre* (or following *Cpos*) segment identity and the syllabic position *Syl* were hidden.

The duration D parent set $\mathbf{Pa}(D)$ of the CBN3 model contains all variables except the within-utterance position *Utt*. We expected that any of the 6 *HIDDEN* variable conditions (Table 6.4) would result in correlation decrease and an increase in the RMS error. However, the CBN3 model predicted consonant duration with significantly lower correlation and higher RMS error only when the following segment *Cpos* identity variable was hidden. As can be seen from Figure F.3 on page 155, the parent set of the *Cpos*

contains 5 variables: $\mathbf{Pa} = \{ Wpos,\ S,\ Utt,\ Syl,\ Cpre \}$. Therefore, there are 5 serial connections between duration D, *Cpos*, and its parents. If *Cpos* is hidden, the duration D and any of the *Wpos, S, Utt, Syl, Cpre* variables are conditionally dependent, i.e. the evidence on any of the *Wpos, S, Utt, Syl, Cpre* variables will influence the duration D variable. We may conclude that the performance of the CBN3 model degrades significantly when the following segment identity *Cpos* is hidden, since the stress S, the within-word position *Wpos*, the within-utterance position *Utt*, and the previous segment identity *Cpre* variables still implicitly influence consonant duration.

The duration D parent set $\mathbf{Pa}(D)$ of the CBN4 model contains all variables except the frontness *Front*. The duration D parent set $\mathbf{Pa}(D)$ of the CBN5 model consists of all 8 linguistic variables. As was expected, both CBN4 and CBN5 models trained under any of the 6 *HIDDEN* variables conditions predicted consonant duration with significantly lower correlation and higher RMS error values than those for the *FULL* observation condition.

The duration D parent set $\mathbf{Pa}(D)$ of the CBN6 model contains all variables except the stress S, the within-utterance position *Utt* and the frontness *Front* (Figure F.5 on page 157). There are 2 diverging connections between the duration D, the within-word position *Wpos*, and either the stress S or the within-utterance position *Utt* variables: $D \leftarrow Wpos \rightarrow S$ and $D \leftarrow Wpos \rightarrow Utt$. When the within-word position Wpos variable is hidden, the evidence on either the stress S or the within-utterance position Utt variables would affect the duration D. Thus, as expected when the position of the consonant within the word *Wpos* was hidden, the CBN6 model predicted consonant duration with significantly lower correlation (higher RMS error) compared to the *FULL* observation condition. However, no other *HIDDEN* variables conditions resulted in any significant loss in the model's performance.

The duration D variable parent set $\mathbf{Pa}(D)$ of the CBN7 model contains all variables except the stress S and the within-utterance position *Utt*. We expected the CBN7 model would behave similarly to the CBN6 model, since

the only difference between the two was an additional parent (of the duration D variable), the frontness *Front* variable. Therefore, the reasoning applied to the CBN6 model is valid for the CBN7 model, and we expected that training the CBN7 model under any of the *HIDDEN* variables conditions would result in the loss of the model's performance. As it turned out, the model's training with the within-word position Wpos, the previous segment identity Cpre, and the pair *Syl* - *Cpos* being hidden, indeed, resulted in significant changes in the model's performance.

The duration parent set $\mathbf{Pa}(D)$ of the CBN8 model contains all variables except the stress S. We expected that any of the 6 *HIDDEN* variables conditions will results in the models performance degradation. However, the CBN8 model did not perform any worse compared to the *FULL* observation condition.

6.6 Choosing the best *MV-compound* model

Given the results of the training under the *FULL* and *HIDDEN* variables conditions (that are presented in full detail in Tables G.1-G.2 of the Appendix), we chose our best models for the RP *lja, rjs* and GA *erm* voices.

For the RP *lja, rjs* voices we designated the CBN1 and CBN3 models as the best models. In terms of the correlation, they all perform better than the SoP model and no worse than the CART model. For example, the CBN3 model predicts consonant duration with the correlation of 0.84 when trained under *FULL* observation condition, with the values ranging from 0.69 to 0.79 when trained under the 6 *HIDDEN* variables conditions. This compares favourably against the values of 0.73 and 0.78 for the SoP and CART models, respectively.

For the GA *erm* voice we selected the CBN4 and CBN6 models as the best models. For example, the CBN4 model predicts consonant duration with the correlation 0.80 when trained under *FULL* observation condition, with the values ranging from 0.54 to 0.80 when trained under the 6 *HIDDEN*

variables conditions. In terms of the correlation, both models perform better (e.g. 0.80 for the CBN4 model) than the SoP (0.76) and no worse than the CART model (0.82). Moreover, in terms of the RMS error, all our models trained under *FULL* and *HIDDEN* variables conditions beat both the SoP and CART models.

When trained under the *HIDDEN* variables the CBN1, CBN3 and CBN6 models turned out to be non-sensitive to the information about the consonant's syllabic position. This result could be used in improving the performance of TTS systems in certain situations. For example, our best models would predict duration of an ambisyllabic consonant[1] with the correlations of at least 0.80, and the RMS error no higher than 4.6ms. Hence, if the information about the syllabic position is missing (hidden), these models again would predict consonant duration with the high correlation and low RMS error.

6.7 Summary of the results

In this chapter, we considered Bayesian models for predicting consonant duration. In our preliminary studies we considered the simple model, CBN1-6, that described a consonant based on its identity, the position within the word, the stress and the frontness of the syllabic vowel, and the number of syllables in the word containing the target consonant. In terms of the RMS error, the CBN1-6 model performed better than both the SoP and CART models. However, it significantly underperformed compared to the CART model, and hence the need for a better robust model arose.

We therefore considered additional linguistic factors that are known to influence consonant duration: the position of the word containing the target consonant within an utterance, the consonant's previous and following segment identity, and its syllabic position. We also represented the consonant

[1] For example, the /*d*/ in the word *ladder* belongs to both the coda of the first syllable, and the onset of the second syllable.

identity as a manner-voice compound variable. Hence, we came up with the 9-variable domain set $\mathbf{U}_9$ consisting of manner-voice *MV*, the within-word position *Wpos*, the stress of the syllabic vowel *S*, the within-utterance position *Utt*, the syllabic position *Syl*, the previous (following) segment identity *Cpos* (*Cpre*), and the frontness of the syllabic vowel *Front*. After we defined the domain set, we performed the network structure search using the K2 structure learning algorithm. We applied the search algorithm to the data, whereby the consonant durations were discretised, since the K2 algorithm can be applied to discrete-valued data only. We identified 8 unique networks (models) that we called the *MV-compound* models. We trained these models under the *FULL* observation condition (all linguistic variables observed) and 6 *HIDDEN* variables conditions.

Model	Voice					
	lja	rjs	erm	lja	rjs	erm
CBN1-6	0.699			2.8		
SoP-vowels-6	0.484			27.2		
CBN1	0.80	0.77	0.70	3.8	4.4	3.8
CBN3	**0.84**	**0.80**	0.70	**3.5**	**4.1**	3.6
CBN4	0.72	0.74	**0.80**	4.6	5.1	4.5
CBN6	0.80	0.74	0.75	4.6	5.2	4.6
SoP	0.73	0.79	0.76	25	26	33
CART	0.78	0.80	0.82	21	20	24

Table 6.7: The best *MV-compound* models. The correlation and RMS error results by model type by voice. *FULL* observation condition.

Table 6.7 shows the correlation and RMS error results for the CBN1-6, the *MV-compound* models trained under the *FULL* observation condition. The results are compared against the SoP and CART models. As can be seen from the summary table, all models perform better than both the SoP and CART models in terms of the RMS error. In terms of the correlation, the CBN3 model for the RP English voices (*lja, rjs*) performs better than the SoP model and no worse than the CART model. For the GA English

voice *erm* the CBN4 model performs better than the SoP and no worse than the CART models.

We also trained the *MV-compound* models under various *HIDDEN* variables conditions. We performed paired t-tests in order to quantify the effect of the state (hidden or observed) of the linguistic variables on the models' performance. Table 6.8 shows the paired t-test results for the 3 best *MV-compound* models. When trained under the *HIDDEN* variables conditions these models predict consonant duration with significantly lower correlation and higher RMS error only when the following segment context or the within-word position of the target consonant were hidden. However, the models perform without any loss in performance when the consonant's syllabic position is not known.

Model	HIDDEN					
	Wpos	Syl	Cpre	Cpos	Syl-Cpre	Syl-Cpos
CBN1				√		
CBN3				√		
CBN4	√	√	√	√	√	√
CBN6	√					

Table 6.8: The paired t-test results for the correlation (RMS error). The best *MV-compound* models. The t-test significant pairs are marked with a tick √. *MV-compound* models.

In general, we may conclude that the Bayesian models based on manner-voice representation of the consonant identity outperform both the sums-of-products and CART models in terms of the RMS error. In terms of the correlation, our best models, i.e the CBN1, CBN3, CBN4, and CBN6, perform better than the sums-of-products model and no worse than the CART model. These models therefore, can be successfully implemented in any real TTS system.

Chapter 7

Conclusions and Future Work

In the introduction to this thesis we briefly discussed the various models for predicting phone duration from rule-based to CART to sums-of-products models. As a motivation for this thesis we specified the desire to explore new approaches to modelling phone duration, with Bayesian models being an example of such new approach. As was specified in the introduction chapter on page 1, Bayesian models have many advantages, which we hope the results presented in this thesis have demonstrated.

In Section 7.1 we will review the most prominent results in the light of the advantages of the Bayesian approach. We also point out some limitations of the approach in Section 7.2. In Section 7.3 we elaborate on how the approach can be extended in the future.

7.1 Highlighted results

7.1.1 Representation of problem domain

One of the benefits of the Bayesian approach is that it allows us to directly represent information about the problem domain. In this particular instance, phone duration is modelled with a hybrid Bayesian network

Bayesian: *FHLR* models						
Model	**Correlation**			**RMSE** (ms)		
	lja	**rjs**	**erm**	**lja**	**rjs**	**erm**
VBN3-10	**0.884**	**0.894**	0.684	2.48	2.35	2.69
VBN4-10	0.857	0.823	0.748	**1.52**	**1.45**	**1.68**
VBN5-10	0.827	0.841	**0.775**	1.85	1.61	1.79
Bayesian: *FH-compound* models						
VBN2-8	0.843	0.836	0.812	**2.81**	**1.61**	**2.75**
VBN3-8	0.828	0.816	0.756	4.0	2.51	4.17
VBN4-8	0.842	**0.843**	0.796	4.1	2.4	4.65
SoP	0.71	0.72	0.70	34.5	37.8	42.1
CART	0.861	0.88	0.89	25.7	23	26.9

Table 7.1: Vowels. The correlation and RMS error results by voice by model type: Bayesian, SoP, and CART models. *FULL* observation condition. The maximum (across different models) values are shown in **boldface**.

consisting of discrete variables for the linguistic factors influencing phone duration, and a continuous variable for the phone duration. Based on the substantial body of linguistic research (discussed in Chapter 3 on pages 20, 24), we selected a set of linguistic factors that are known to influence phone duration.

For example, for vowels we had two models: one was based on a 2-feature (*FHLR* model) and the other on a 4-feature (*FH-compound* model) vowel identity representation. For the *FHLR* models we represented vowel identity using a set of 4 variables corresponding to the frontness, height, length, and roundness distinctive features. As can be seen from Table 7.1 the best *FHLR* models trained under the *FULL* observation condition performed with correlations ranging from 0.823 to 0.894 for the RP voices (*rjs,lja*), and from 0.60 to 0.775 for the GA voice (*erm*). In terms of the correlation and RMS error, the *FHLR* models all (except for the VBN7-10 model for the *erm* voice) significantly outperform the SoP models and perform no worse than the CART models, as can be seen from Table 5.2 on page 82.

As one can see from Table 7.1, the best *FH-compound* models trained

under the *FULL* observation condition predicted vowel durations of the test set with correlations ranging from 0.816 to 0.844 and from 0.756 to 0.812 for the RP *rjs,lja* voices and GA voice, respectively. However, in terms of the RMS error all models in these two classes outperformed both the SoP and CART models. These results were slightly worse than those for the *FHLR* models. One may also notice from Table 5.1 on page 79 that the best *FHLR* models for the RP voices, the VBN3-10 and VBN4-10 models, had the frontness *Front*, height *Height*, and length *Length* as the parents of duration D, whereas the best model for the GA *erm* voice, the VBN5-10 model, had the following segment identity *Cpos* and the roundness *Rnd* variables as the parents of duration D. This result may imply that the choice of variables for duration parent set $\mathbf{Pa}(D)$ is dialect dependent. These results also signify the importance of the *Length* variable for predicting vowel duration, since 6 out of 7 duration parent sets $\mathbf{Pa}(D)$ contained the length *Length* variable.

Bayesian: *MV-compound* models						
Model	**Correlation**			**RMSE** (ms)		
	lja	**rjs**	**erm**	**lja**	**rjs**	**erm**
CBN1-6		0.699			2.8	
CBN1	0.80	0.77	0.69	3.8	4.4	3.8
CBN3	**0.84**	**0.80**	0.69	**3.5**	**4.1**	3.6
CBN4	0.72	0.74	**0.80**	4.6	5.1	4.5
CBN6	0.80	0.74	0.75	4.6	5.2	4.6
SoP	0.73	0.79	0.76	25	26	33
CART	0.78	0.80	0.82	21	20	24

Table 7.2: Consonants. The best *MV-compound* models. The correlation and RMS error results by voice by model type: Bayesian, SoP, and CART models. *FULL* observation condition.

For consonants, we initially devised a model (CBN1-6), whereby we represented consonant identity as an enumerative consonant type variable. In terms of the correlation, the model performed better than the SoP model, but worse than the CART model, as can be seen from Table 6.2 on page

102. We needed a better, more robust model that would beat both the SoP and CART models. We defined a new problem domain, whereby we represented the consonant identity as a manner-voice compound variable. As can be seen from Table 6.7 on page 116, the *MV-compound* models based on this representation all outperform the SoP and CART models in terms of the RMS error. For the *lja* and *rjs* voices, in terms of the correlation, the CBN1, CBN3, and CBN6 models perform better than the SoP models, and no worse than the CART model. For the *erm* voice, the CBN4 model outperforms the SoP model and is comparable in performance to the CART model.

Therefore, the linguistically-rooted phone identity representation as well as the use of other linguistically motivated variables such as lexical stress, within-word position, among others, result in better, more robust performance of the models.

7.1.2 Model structure and factor interaction

Bayesian models, being a special case of graphical models, allow for a compact intuitive representation of the variables' (factor) interaction. Moreover, they allows for an explicit representation of the causal dependencies among the problem domain variables. Causal dependencies as well as variable interaction in the model are represented by edges in the corresponding DAG, and the joint probability distribution $P(\mathbf{x})$ (JPD) over the model's variables quantifies these dependency relations among the problem domain variables.

Bayesian models are constructed using the expert knowledge or machine learning techniques. In a pilot study, we devised a model structure by hand. Consequently, the variable interactions were very much dependent on the amount of expert knowledge embedded in the network structure of the model. Overall, our hand-picked model performed worse than the SoP and CART models. Therefore, we took a machine learning approach to learning the model structure (i.e. the relations among the variables) in the

hope to improve the model's performance. To learn the model's structure we applied the K2 structure learning algorithm to our data, in which the duration values were discretised, since the K2 structure learning algorithm is defined for purely discrete networks. After we learnt the structures, we switched back to the original, continuously-valued duration data. From the

# orderings	# unique DAGs	$\mathbf{Pa}(D)$ equivalent
40,320	947	7
5,040	590	4
40,320	910	8

Table 7.3: The total number of variable orderings, the number of unique DAGs learnt by the K2 algorithm, and the number of duration parent set equivalent DAGs.

DAGs learnt, we pre-selected unique ones based on the equality of the corresponding adjacency matrices. Following this, we selected unique DAG structures based on the duration parent equivalence assumption discussed in Chapter 5 on page 77 (the DAGs that have the same duration parent set $\mathbf{Pa}(D)$ are assumed to be equivalent). In this way we identified 7, 4, and 8 different network topologies for the *FHLR*, *FH-compound*, and *MV-compound* models, respectively. (Table 7.3 shows the information about the learnt DAGs.)

Our machine-learnt models turned out to perform better than the hand-crafted model. For example, our hand-crafted model CBN1-6 predicted consonant duration with a correlation of 0.69 compared to the 0.80 of the CBN3 model, whose structure was learnt from data. Overall, all (except for the CBN8 model) machine-learnt models performed better than the hand-crafted model. This result clearly demonstrates the advantages of models learnt from data using a machine learning technique (i.e. the K2 structure learning algorithm) over hand-picked models.

7.1.3 Prediction in case of missing (hidden) data

Vowels

k	HIDDEN	Model			
		VBN1-8	VBN2-8	VBN3-8	VBN4-8
7	Cpos			√	
8	Cpos Wd			√	
9	Cpos Rnd			√	
10	Cpos FH			√	
15	Utt Cpos	√	√	√	
16	S				√
17	S Wd				√
18	S Rnd			√	
21	S Utt			√	
26	Wpos Cpos	√	√	√	
28	Wpos S				√

Table 7.4: Vowels: *FH-compound* networks. The *HIDDEN* variables conditions that resulted in significant ($p < 0.01$, one-tailed) decrease in the correlation and increase in the RMS error are marked with a tick √. The *HIDDEN* variables condition names are shown in Table 5.6 on page 87.

Bayesian models can make robust predictions in cases of missing (hidden) data. Overall, the performance of the *FH-compound* models trained under the *HIDDEN* variables condition degraded only in fewer than 50% (12 out of 28) of the hidden conditions. Table 7.4 shows only those *HIDDEN* conditions that resulted in a reduction of the models' performance.

When experimenting with different *HIDDEN* variables conditions, we can deduce the most important variables for predicting phone duration. In particular, the following segment context as well as the word- and utterance-level context turned out to be the most important predictors of vowel duration. For example, the VBN1-8, VBN2-8, and VBN3-8 models predicted vowel duration with significantly lower correlation and higher RMS error only when the following segment identity *Cpos* and the within-word position

Wpos, or when the following segment identity *Cpos* and the within-utterance position *Utt* were hidden. In addition, the roundness and the stress factors were found to be equally important for predicting vowel duration. For the VBN3-8 model, it turned out that the stress S and the following segment identity *Cpos* are the most important predictors of vowel duration. When the stress S and any other linguistic variable (except *Cpos*) were hidden, the model predicted vowel duration with significantly lower correlation (higher RMS error) values than these for the *FULL* observation condition. Likewise, when the following segment identity *Cpos* and any other linguistic variable (except S) were hidden the model's performance was significantly worse than that for the *FULL* observation condition.

As can be seen from Table 7.4, for 2 sparsely connected models, i.e. the VBN1-8 and VBN2-8, there were only 3 factors out of those 8 discussed in the literature reviewed (Section 3.1 on page 20) that significantly affected the Bayesian duration prediction. These were the following segment identity *Cpos*, the within-word *Wpos* and within-utterance *Utt* position factors. For densely connected models, i.e. the VBN3-8 and VBN4-8, there were also stress S, vowel identity (represented as the front-height *FH* and roundness *Rnd*) and word class *Wd* factors that significantly influenced vowel duration.

However, in the majority of the *HIDDEN* conditions, all models predicted vowel duration with correlation and RMS error values no worse than those for the *FULL* observation condition. Moreover, the best models trained under the *HIDDEN* variables conditions still performed better than the SoP model, and no worse than the CART model, as can be seen from Table 5.5 on page 86 and Tables E.1-E.4 on pages 148-151 of the Appendix.

Consonants

Our Bayesian models for predicting consonant duration demonstrated quite robust behaviour when trained under the 6 *HIDDEN* variables conditions. As can be seen from Table 7.5 the performance of the models does not degrade much compared to the *FULL* observation condition, except for the

		Model							
k	*HIDDEN*	CBN1	CBN2	CBN3	CBN4	CBN5	CBN6	CBN7	CBN8
1	Wpos				√	√	√	√	
2	Syl				√	√			
3	Cpre				√	√		√	
4	Cpos	√		√	√	√			
5	Syl-Cpre				√	√			
6	Syl-Cpos				√	√		√	

Table 7.5: Consonants: *MV-compound* networks. The *HIDDEN* variables conditions that resulted in significant ($p < 0.01$, one-tailed) decrease in the correlation (increase in the RMS error) are marked with a tick √. The *HIDDEN* variables condition names are shown in Table 6.4 on page 106.

CBN4 (see Figure 6.3 on page 104) and CBN5 (Figure F.4 on page 156) models that performed significantly worse under all HIDDEN variables conditions.

As can also be seen from Table 7.5, there were 2 factors (see literature review of Section 3.2 on page 24) that significantly influenced consonant duration prediction in 50% of the models. These were the following segment identity *Cpos* and the within-word position of the target consonant *Wpos* factors. In addition, the previous segment identity *Cpre* factor was found to significantly affect consonant duration prediction in 38% of the models.

Overall, the CBN1, CBN3, and CBN6 models performed better than the SoP models, and no worse than the CART model. They predicted consonant duration with significantly ($p < 0.01$, one-tailed) lower correlation and higher RMS error values only when the following segment identity *Cpos* or the within-word position of the target consonant *Wpos* were hidden. To put it another way, the following context as well as the word level information turned out to be very important for consonant duration prediction using these models. However, the syllabic position *Syl* turned out to be unimportant in predicting consonant duration using these models. Thus, if the syllabic position *Syl* is hidden or missing, we expect that our best models

will predict consonant duration just as well as if *Syl* is observed.

Experimenting with various linguistic variables being hidden during training allowed us to investigate the models' robustness. These experiments also allowed us to identify the linguistic factors that play a crucial role in predicting phone duration.

7.2 Limitations of the approach

7.2.1 Problem domain specification

Despite being successful overall, there were some limitations to predicting phone duration using Bayesian models. It turns out that for the GA *erm* voice, the models learnt from data demonstrated slightly low performance in comparison to the results for the 2 RP voices: *lja, rjs*. For example, the CBN3 model predicted consonant duration with a correlation of 0.69 for the GA *erm* voice, and with correlations of 0.84 and 0.80 for the RP *lja* and *rjs* voices, respectively, as can be seen from Table 7.2.

Such a difference can not be explained by the training data size, since the training set for the *lja* voice contains 54,489 consonant tokens, and for the *erm* voice 85,048 tokens. This tendency of the models under performing for the *erm* voice in comparison to the *lja, rjs* voices can be observed across different models for both vowels and consonants (see Table 5.5 on page 86 and Table 6.5 on page 108).

One possible explanation for such under performance of the models trained on the *erm* voice is the choice of the linguistic problem domain variables. It may well be that 4 distinctive features suffice to describe the RP but not the GA vowels. For GA vowels we may need extra features. For example, we could introduce a variable *Tense* for the distinctive feature tenseness that would describe the articulatory effort at the qualitative level during pronouncing a vowel sound. In addition, since GA is a rhotic accent of English, we may need to introduce a variable to acknowledge this fact.

7.2.2 Bayesian parameters

A few words need to be said about the parameter estimation. First of all, the number of parameters of duration D variable is exponential in the cardinality of its parents. Given that all variables are multi-valued, the number of the parameters can be quite large. For example, for the CBN5 model for consonants the number of parameters for the duration D variable is $13,122$, which is almost 70 times larger than the number of 196 reported in van Santen (1994) for SoP models for consonants.

The parameter estimation using the EM algorithm is the most time-consuming part of the model training process. On average, it takes from 2 to 48 hours to train the BN model of size 8. In the future, to cut down on the training time, we can specify the parameters in more optimal way: instead of storing a full table we can store fewer parameters using standard techniques such as noisy-OR Pearl (1988), decision trees Boutilier, Friedman, Goldszmidt & Koller (1996) or default tables Friedman & Goldszmidt (1996).

7.3 Future work

7.3.1 Experimenting with new features for the problem domain

When discussing the limitation of the current approach, we mentioned the necessity of using additional features for describing a particular dialect. In the future we will experiment with the place of articulation feature for consonants. For vowels, we will introduce the tenseness feature for GA English.

7.3.2 Model structure learning

We applied the K2 structure algorithm to find all possible model structures. Due to time limitation we could not perform a search of the whole space of model structures: for all variable orderings we searched just one discretisa-

tion slice. Performing a selective search, we may have missed some of the structures. Time permitting, we will perform the search through the whole space of hypotheses, running the K2 algorithm for 9 discretisation levels.

7.3.3 Building models for a new data set

Once we have built the model for a particular data set (call it *source*), we can apply it to build a model for a new data set (*target*). We essentially can follow a few different approaches. First, we can use the complete models (structure and parameters) learnt from the source data to perform the inference on the target data set. Second, we can use the model structure learnt from the source data to learn the target model parameters. However, if the target data set is small, we would select the most significant (based on the source data set inference results) factors first, and then perform structure and parameter learning for the target data.

7.4 Selective training

When we trained the models under *HIDDEN* variables condition, we used a fixed set of hidden variables. Due to the properties of the Bayesian models, some of these hidden conditions were redundant, i.e. we did not expect any changes in the model performance under these redundant conditions. In order to study the effect of the hidden state of a particular variable or a set of variables on the model's performance, we should instead, define separate sets of *HIDDEN* variables conditions for each model based on its structure.

7.4.1 Model extension

Bayesian models are easily extensible. Therefore, we can improve models' performance by choosing a different BN topology for each phone type. The straightforward approach would be: out of a set of models learnt from data for each phone type, choose a model that predicts a phone duration with the maximum correlation and minimum RMS error. We can organise these

models into a classification tree based on a set of linguistic features such as frontness, height etc. for vowels, and place or manner of articulation, or syllabic position for consonants. We choose appropriate model by classifying the phone starting from the root feature and going down to the pre-terminal leaf. The leaf of the tree is the appropriate Bayesian model.

7.5 Conclusions

The results presented in this thesis have demonstrated the following: Bayesian models allow for an intuitive, straightforward representation of the problem domain information. The proper problem domain representation does matter for the performance of the model. Bayesian models allow for easy experimentation with different problem domain representations, thus allowing the model designer to choose the best model. The models presented, explicitly represent linguistic factor interaction. In particular, training the models with some of the linguist variables being hidden, allowed us to investigate the relative importance of linguistic factors for predicting phone duration. Our models made robust predictions in cases of hidden and missing data. The models' structure as well as parameters are easily estimated from the data. These models therefore, can be successfully implemented in any real TTS system. Building and training a model may be a time consuming process. But once the model is built and trained, it is computationally cheap to use for duration prediction, since it is essentially a look-up table, if all parents are observed. The model therefore, can be successfully implemented in any real TTS system.

Appendix A

FHLR networks learnt by the K2 algorithm

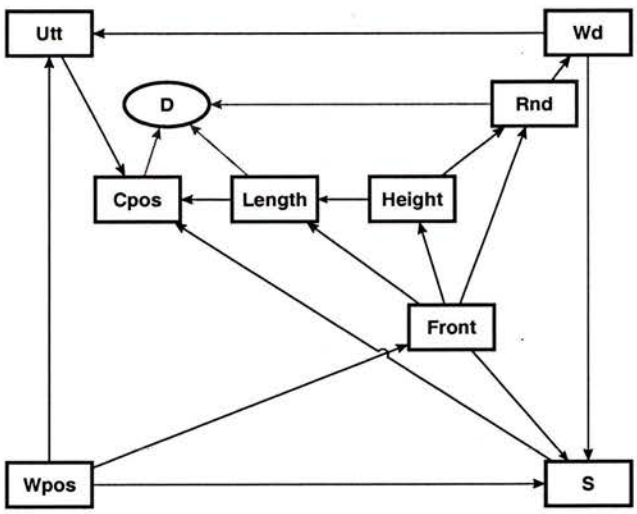

Figure A.1: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN1-10 model.

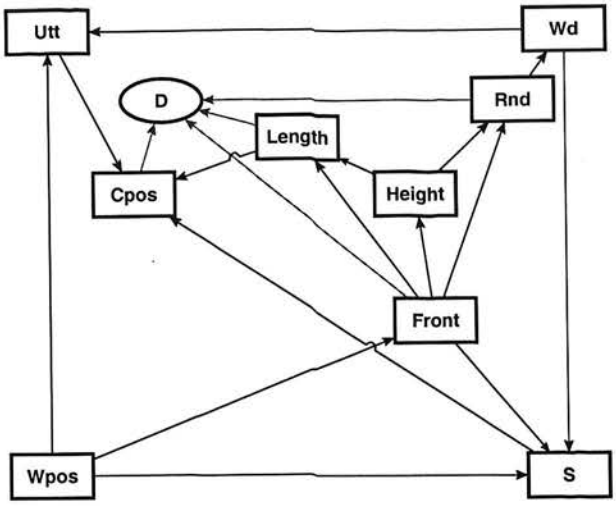

Figure A.2: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN2-10 model.

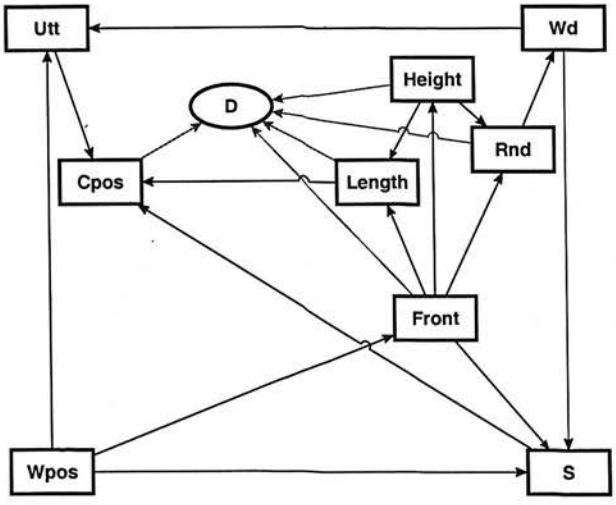

Figure A.3: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN3-10 model.

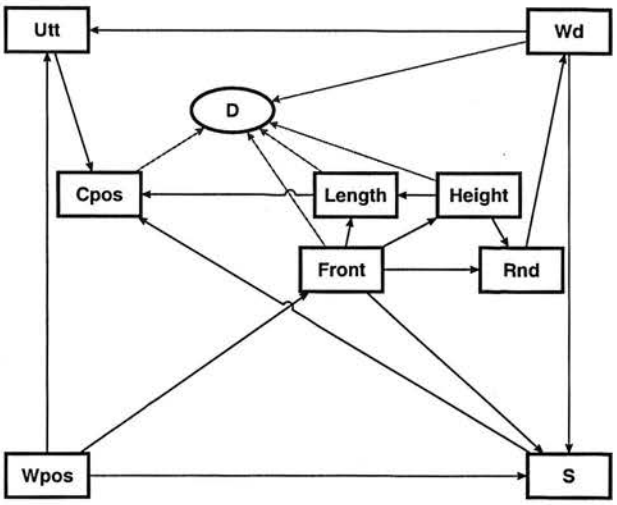

Figure A.4: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN4-10 model.

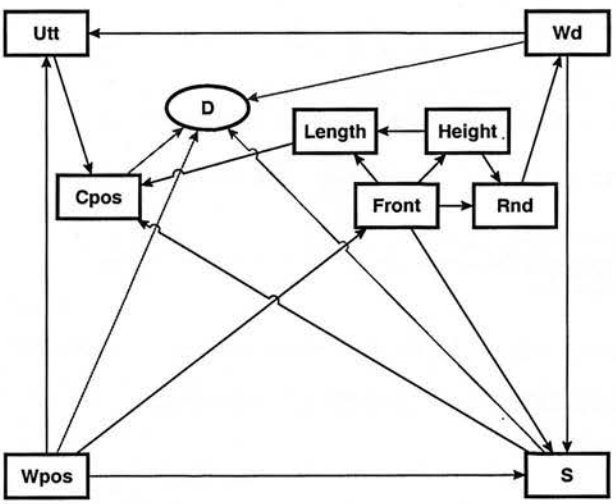

Figure A.5: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN5-10 model.

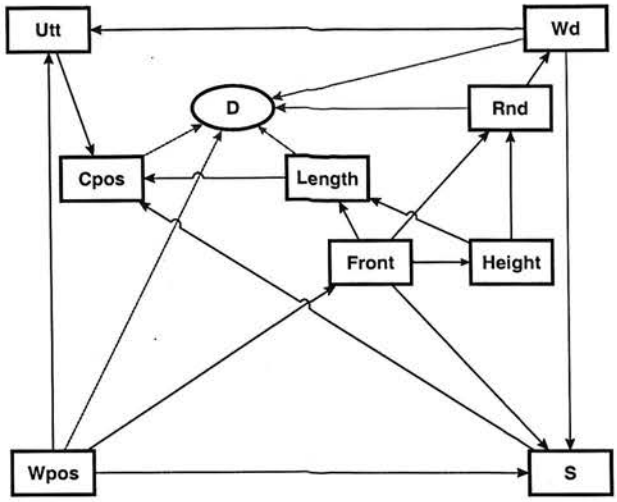

Figure A.6: The *FHLR* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN6-10 model.

Appendix B

FH-compound networks learnt by the K2 algorithm

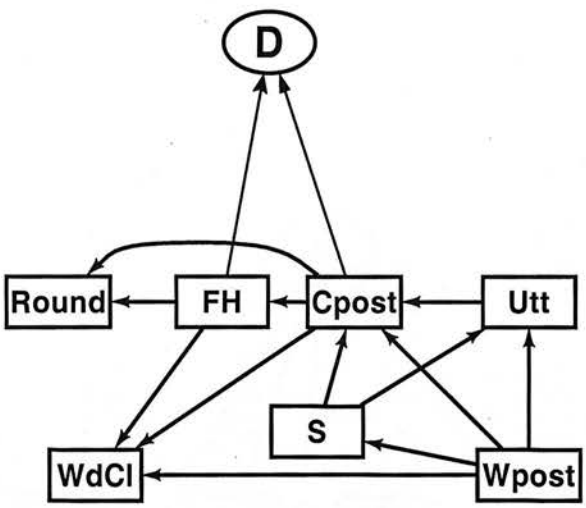

Figure B.1: The *FH-compound* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN1-8 model.

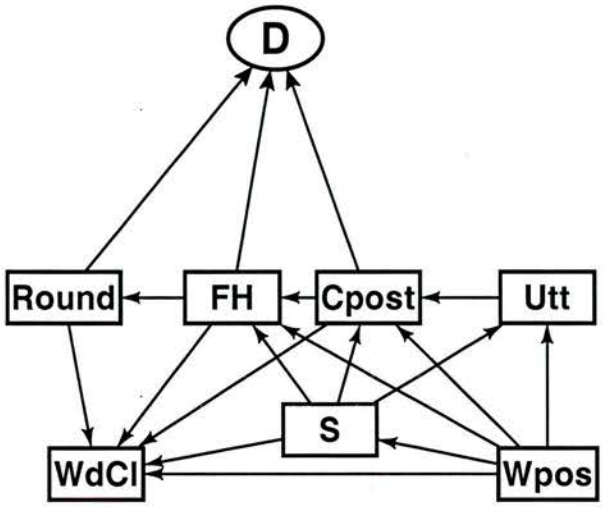

Figure B.2: The *FH-compound* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN2-8 model.

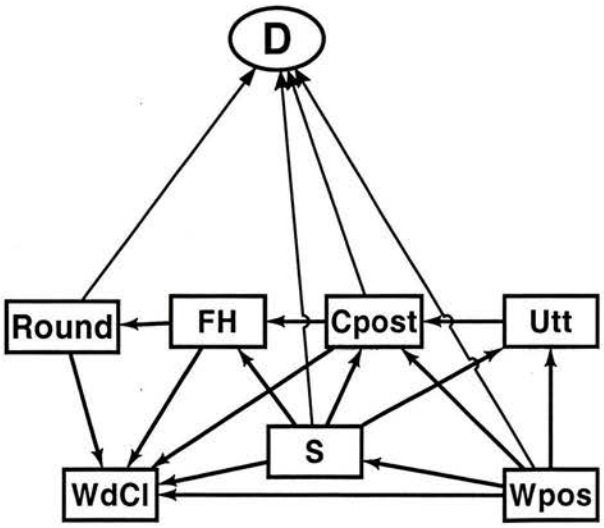

Figure B.3: The *FH-compound* network learned by the K2 algorithm, with vowel durations being uniformly discretised. The VBN3-8 model.

Appendix C

FH-compound networks: correlation results

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
4	FH	0.052	0.034	0.020	-0.033	0.137	2.641	2	0.118
5	FH Wd	0.047	0.027	0.015	-0.019	0.114	3.094	2	0.091
6	FH Rnd	0.054	0.031	0.018	-0.023	0.130	3.032	2	0.094
7	Cpos	0.161	0.123	0.071	-0.146	0.467	2.258	2	0.153
8	Cpos Wd	0.170	0.113	0.065	-0.112	0.451	2.591	2	0.122
9	Cpos Rnd	0.172	0.140	0.081	-0.176	0.520	2.122	2	0.168
10	Cpos FH	0.037	0.021	0.012	-0.014	0.089	3.124	2	0.089
14	Utt FH	0.052	0.034	0.020	-0.033	0.137	2.641	2	0.118
15	Utt Cpos	0.178	0.121	0.070	-0.123	0.480	2.547	2	0.126
19	S FH	0.052	0.034	0.020	-0.033	0.137	2.641	2	0.118
20	S Cpos	0.179	0.121	0.070	-0.122	0.479	2.557	2	0.125
25	Wpos FH	0.061	0.045	0.026	-0.050	0.172	2.365	2	0.142
26	Wpos Cpos	0.181	0.140	0.081	-0.166	0.529	2.242	2	0.154

Table C.1: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *FH-compound* networks: The VBN1-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wd	-0.002	0.004	0.002	-0.013	0.008	-1.000	2	0.423
2	Rnd	0.023	0.010	0.006	-0.002	0.048	4.020	2	0.057
3	Rnd Wd	0.029	0.020	0.012	-0.021	0.078	2.471	2	0.132
4	FH	0.026	0.012	0.007	-0.004	0.056	3.768	2	0.064
5	FH Wd	0.054	0.036	0.021	-0.035	0.144	2.607	2	0.121
6	FH Rnd	0.022	0.012	0.007	-0.010	0.053	2.981	2	0.097
7	Cpos	0.119	0.059	0.034	-0.028	0.266	3.489	2	0.073
8	Cpos Wd	0.126	0.063	0.036	-0.031	0.282	3.448	2	0.075
9	Cpos Rnd	0.075	0.069	0.040	-0.097	0.248	1.880	2	0.201
10	Cpos FH	0.051	0.038	0.022	-0.043	0.146	2.333	2	0.145
11	Utt	-0.002	0.004	0.002	-0.013	0.008	-1.000	2	0.423
13	Utt Rnd	0.023	0.010	0.006	-0.002	0.048	4.020	2	0.057
14	Utt FH	0.026	0.012	0.007	-0.004	0.056	3.768	2	0.064
15	Utt Cpos	0.116	0.0291	0.0168	0.044	0.188	6.91	2	0.020
18	S Rnd	0.024	0.014	0.008	-0.010	0.058	3.060	2	0.092
19	S FH	0.028	0.016	0.009	-0.011	0.068	3.061	2	0.092
20	S Cpos	0.112	0.070	0.040	-0.062	0.285	2.775	2	0.109
24	Wpos Rnd	0.022	0.011	0.007	-0.006	0.050	3.388	2	0.077
25	Wpos FH	0.047	0.043	0.025	-0.059	0.153	1.925	2	0.194
26	Wpos Cpos	0.117	0.026	0.015	0.051	0.181	7.686	2	0.017

Table C.2: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *FH-compound* networks: The VBN2-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
2	Rnd	0.021	0.015	0.008	-0.015	0.057	2.530	2	0.127
3	Rnd Wd	0.022	0.015	0.009	-0.015	0.059	2.592	2	0.122
6	FH Rnd	0.023	0.018	0.011	-0.023	0.068	2.133	2	0.167
7	Cpos	0.092	0.037	0.021	0.001	0.184	4.330	2	0.049
8	Cpos Wd	0.096	0.037	0.022	0.003	0.189	4.437	2	0.047
9	Cpos Rnd	0.100	0.023	0.013	0.042	0.157	7.483	2	0.017
10	Cpos FH	0.089	0.037	0.022	-0.004	0.182	4.138	2	0.054
13	Utt Rnd	0.021	0.015	0.008	-0.015	0.057	2.530	2	0.127
15	Utt Cpos	0.070	0.033	0.019	-0.013	0.153	3.618	2	0.069
16	S	0.052	0.030	0.017	-0.022	0.126	3.024	2	0.094
17	S Wd	0.055	0.024	0.014	-0.006	0.115	3.873	2	0.061
18	S Rnd	0.06461	0.01610	0.00929	0.02462	0.10459	6.952	2	0.020
19	S FH	0.056	0.027	0.016	-0.012	0.123	3.525	2	0.072
20	S Cpos	0.107	0.041	0.023	0.006	0.208	4.554	2	0.045
21	S Utt	0.046	0.032	0.018	-0.033	0.125	2.481	2	0.131
22	Wpos	0.050	0.041	0.023	-0.052	0.151	2.110	2	0.169
23	Wpos Wd	0.047	0.048	0.028	-0.071	0.166	1.717	2	0.228
24	Wpos Rnd	0.060	0.036	0.021	-0.028	0.148	2.933	2	0.099
25	Wpos FH	0.046	0.046	0.027	-0.068	0.161	1.750	2	0.222
26	Wpos Cpos	0.095	0.038	0.022	0.0001	0.190	4.289	2	0.050
27	Wpos Utt	0.040	0.046	0.026	-0.073	0.154	1.534	2	0.265
28	Wpos S	0.074	0.054	0.031	-0.061	0.209	2.371	2	0.141

Table C.3: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *FH-compound* networks: The VBN3-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wd	0.040	0.016	0.009	0.0001	0.080	4.265	2	0.051
2	Rnd	0.042	0.028	0.016	-0.029	0.113	2.564	2	0.124
3	Rnd Wd	0.063	0.040	0.023	-0.037	0.164	2.720	2	0.113
4	FH	0.105	0.103	0.059	-0.151	0.361	1.768	2	0.219
5	FH Wd	0.127	0.121	0.070	-0.173	0.428	1.826	2	0.209
6	FH Rnd	0.131	0.126	0.073	-0.182	0.445	1.803	2	0.213
7	Cpos	0.149	0.103	0.059	-0.106	0.404	2.520	2	0.128
8	Cpos Wd	0.163	0.128	0.074	-0.155	0.480	2.201	2	0.159
9	Cpos Rnd	0.150	0.117	0.067	-0.140	0.440	2.223	2	0.156
10	Cpos FH	0.155	0.167	0.096	-0.260	0.569	1.602	2	0.250
11	Utt	0.072	0.018	0.011	0.026	0.118	6.753	2	0.021
12	Utt Wd	0.095	0.027	0.015	0.028	0.161	6.116	2	0.026
13	Utt Rnd	0.133	0.120	0.069	-0.165	0.431	1.924	2	0.194
14	Utt FH	0.170	0.167	0.096	-0.245	0.584	1.761	2	0.220
15	Utt Cpos	0.135	0.110	0.064	-0.139	0.408	2.118	2	0.168
16	S	0.073	0.006	0.004	0.058	0.089	20.842	2	0.002
17	S Wd	0.100	0.018	0.011	0.055	0.146	9.548	2	0.011
18	S Rnd	0.090	0.054	0.031	-0.045	0.225	2.874	2	0.103
19	S FH	0.146	0.136	0.079	-0.192	0.484	1.855	2	0.205
20	S Cpos	0.193	0.136	0.078	-0.145	0.530	2.455	2	0.133
21	S Utt	0.103	0.061	0.035	-0.049	0.255	2.909	2	0.101
22	Wpos	0.105	0.034	0.020	0.020	0.189	5.323	2	0.034
23	Wpos Wd	0.125	0.070	0.041	-0.049	0.300	3.088	2	0.091
24	Wpos Rnd	0.123	0.094	0.054	-0.110	0.356	2.267	2	0.152
25	Wpos FH	0.131	0.107	0.062	-0.135	0.397	2.118	2	0.168
26	Wpos Cpos	0.164	0.135	0.078	-0.171	0.500	2.110	2	0.169
27	Wpos Utt	0.132	0.101	0.058	-0.119	0.383	2.256	2	0.153
28	Wpos S	0.126	0.006	0.003	0.111	0.141	36.002	2	0.001

Table C.4: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *FH-compound* networks: The VBN4-8 model.

Appendix D

FH-compound networks: RMS error results

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wd	-0.002	0.004	0.002	-0.013	0.008	-1.000	2	0.423
2	Rnd	0.023	0.010	0.006	-0.002	0.048	4.020	2	0.057
3	Rnd Wd	0.029	0.020	0.012	-0.021	0.078	2.471	2	0.132
4	FH	0.026	0.012	0.007	-0.004	0.056	3.768	2	0.064
5	FH Wd	0.054	0.036	0.021	-0.035	0.144	2.607	2	0.121
6	FH Rnd	0.022	0.012	0.007	-0.010	0.053	2.981	2	0.097
7	Cpos	0.119	0.059	0.034	-0.028	0.266	3.489	2	0.073
8	Cpos Wd	0.126	0.063	0.036	-0.031	0.282	3.448	2	0.075
9	Cpos Rnd	0.075	0.069	0.040	-0.097	0.248	1.880	2	0.201
10	Cpos FH	0.051	0.038	0.022	-0.043	0.146	2.333	2	0.145
11	Utt	-0.002	0.004	0.002	-0.013	0.008	-1.000	2	0.423
13	Utt Rnd	0.023	0.010	0.006	-0.002	0.048	4.020	2	0.057
14	Utt FH	0.026	0.012	0.007	-0.004	0.056	3.768	2	0.064
15	Utt Cpos	-1.334	0.294	0.170	-2.07	-0.60	-7.846	2	0.016
18	S Rnd	0.024	0.014	0.008	-0.010	0.058	3.060	2	0.092
19	S FH	0.028	0.016	0.009	-0.011	0.068	3.061	2	0.092
20	S Cpos	0.112	0.070	0.040	-0.062	0.285	2.775	2	0.109
24	Wpos Rnd	0.022	0.011	0.007	-0.006	0.050	3.388	2	0.077
25	Wpos FH	0.047	0.043	0.025	-0.059	0.153	1.925	2	0.194
26	Wpos Cpos	-3.045	0.751	0.433	-4.91	-1.178	-7.020	2	.020

Table D.1: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *FH-compound* networks: The VBN1-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wd	0.017	0.030	0.017	-0.057	0.091	1.000	2	0.423
2	Rnd	-0.66	0.163	0.094	-1.062	-0.25	-7.015	2	0.020
3	Rnd Wd	-0.335	0.132	0.076	-0.664	-0.006	-4.383	2	0.048
4	FH	-0.204	0.106	0.061	-0.468	0.060	-3.330	2	0.080
5	FH Wd	-0.428	0.376	0.217	-1.362	0.506	-1.973	2	0.187
6	FH Rnd	-0.194	0.094	0.054	-0.427	0.039	-3.588	2	0.070
7	Cpos	-1.169	0.518	0.299	-2.455	0.117	-3.912	2	0.060
8	Cpos Wd	-1.243	0.521	0.301	-2.536	0.051	-4.134	2	0.054
9	Cpos Rnd	-0.663	0.705	0.407	-2.415	1.089	-1.628	2	0.245
10	Cpos FH	-0.386	0.282	0.163	-1.087	0.315	-2.370	2	0.141
11	Utt	0.017	0.030	0.017	-0.057	0.091	1.000	2	0.423
13	Utt Rnd	-0.309	0.082	0.047	-0.512	-0.106	-6.538	2	0.023
14	Utt FH	-0.204	0.106	0.061	-0.468	0.060	-3.330	2	0.080
15	Utt Cpos	-1.032	0.380	0.220	-1.977	-0.088	-4.702	2	0.042
18	S Rnd	-0.311	0.092	0.053	-0.541	-0.082	-5.831	2	0.028
19	S FH	-0.263	0.164	0.095	-0.670	0.145	-2.774	2	0.109
20	S Cpos	-1.049	0.617	0.356	-2.581	0.482	-2.947	2	0.098
24	Wpos Rnd	-0.299	0.103	0.059	-0.554	-0.044	-5.051	2	0.037
25	Wpos FH	-0.429	0.433	0.250	-1.505	0.647	-1.716	2	0.228
26	Wpos Cpos	-1.076	0.346	0.200	-1.936	-0.216	-5.384	2	0.033

Table D.2: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *FH-compound* networks: The VBN2-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
2	Rnd	-0.281	0.160	0.092	-0.679	0.117	-3.041	2	0.093
3	Rnd Wd	-0.293	0.162	0.094	-0.696	0.110	-3.125	2	0.089
7	FH Rnd	-0.294	0.202	0.117	-0.796	0.208	-2.520	2	0.128
8	Cpos	-1.253	0.113	0.065	-1.534	-0.972	-19.204	2	0.003
9	Cpos Wd	-1.345	0.117	0.067	-1.636	-1.055	-19.932	2	0.003
10	Cpos Rnd	-1.467	0.414	0.239	-2.497	-0.438	-6.133	2	0.026
11	Cpos FH	-1.271	0.219	0.127	-1.816	-0.726	-10.031	2	0.010
14	Utt Rnd	-0.281	0.160	0.092	-0.679	0.117	-3.041	2	0.093
16	Utt Cpos	-0.796	0.010	0.006	-0.822	-0.770	-132.242	2	0.0001
17	S	-0.530	0.197	0.114	-1.018	-0.041	-4.667	2	0.043
18	S Wd	-0.571	0.185	0.107	-1.030	-0.113	-5.358	2	0.033
19	S Rnd	-0.825	0.262	0.151	-1.475	-0.175	-5.464	2	0.032
20	S FH	-0.583	0.197	0.114	-1.073	-0.093	-5.115	2	0.036
21	S Cpos	-1.479	0.315	0.182	-2.262	-0.696	-8.127	2	0.015
22	S Utt	-0.390	0.219	0.127	-0.935	0.154	-3.086	2	0.091
23	Wpos	-0.710	0.258	0.149	-1.350	-0.070	-4.776	2	0.041
24	Wpos Wd	-0.656	0.328	0.189	-1.471	0.159	-3.463	2	0.074
25	Wpos Rnd	-1.018	0.360	0.208	-1.911	-0.125	-4.903	2	0.039
26	Wpos FH	-0.657	0.323	0.187	-1.460	0.146	-3.522	2	0.072
27	Wpos Cpos	-1.555	0.313	0.181	-2.333	-0.778	-8.605	2	0.013
28	Wpos Utt	-0.457	0.300	0.173	-1.201	0.287	-2.642	2	0.118
29	Wpos S	-0.922	0.308	0.178	-1.687	-0.157	-5.186	2	0.035

Table D.3: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *FH-compound* networks: The VBN3-8 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wd	-0.423	0.280	0.162	-1.118	0.273	-2.615	2	0.120
2	Rnd	-0.497	0.301	0.174	-1.246	0.252	-2.856	2	0.104
3	Rnd Wd	-0.725	0.451	0.260	-1.845	0.395	-2.785	2	0.108
4	FH	-1.184	1.047	0.604	-3.785	1.417	-1.959	2	0.189
5	FH Wd	-1.470	1.251	0.723	-4.579	1.638	-2.035	2	0.179
6	FH Rnd	-1.605	1.341	0.774	-4.936	1.726	-2.073	2	0.174
7	Cpos	-1.632	1.117	0.645	-4.406	1.142	-2.532	2	0.127
8	Cpos Wd	-1.730	1.361	0.786	-5.112	1.652	-2.201	2	0.159
9	Cpos Rnd	-0.921	1.034	0.597	-3.489	1.647	-1.543	2	0.263
10	Cpos FH	-2.012	2.173	1.255	-7.411	3.386	-1.604	2	0.250
11	Utt	-0.913	0.331	0.191	-1.734	-0.091	-4.781	2	0.041
12	Utt Wd	-1.127	0.499	0.288	-2.366	0.112	-3.913	2	0.060
13	Utt Rnd	-1.548	1.256	0.725	-4.667	1.572	-2.134	2	0.166
14	Utt FH	-2.157	1.843	1.064	-6.736	2.422	-2.027	2	0.180
15	Utt Cpos	-2.263	2.141	1.236	-7.581	3.055	-1.831	2	0.209
16	S	-0.800	0.308	0.178	-1.564	-0.035	-4.502	2	0.046
17	S Wd	-1.28	0.33	0.19	-2.09	-0.47	-6.789	2	0.021
18	S Rnd	-1.044	0.671	0.387	-2.711	0.623	-2.694	2	0.115
19	S FH	-1.665	1.363	0.787	-5.050	1.721	-2.115	2	0.169
20	S Cpos	-2.061	1.430	0.826	-5.613	1.491	-2.496	2	0.130
21	S Utt	-1.244	0.725	0.419	-3.045	0.558	-2.970	2	0.097
22	Wpos	-1.170	0.448	0.259	-2.282	-0.057	-4.524	2	0.046
23	Wpos Wd	-1.348	0.849	0.490	-3.457	0.762	-2.749	2	0.111
24	Wpos Rnd	-1.524	1.157	0.668	-4.399	1.351	-2.281	2	0.150
25	Wpos FH	-1.589	1.213	0.701	-4.603	1.426	-2.268	2	0.151
26	Wpos Cpos	-1.862	1.467	0.847	-5.507	1.782	-2.199	2	0.159
27	Wpos Utt	-1.547	1.146	0.661	-4.393	1.299	-2.339	2	0.144
28	Wpos S	-1.432	0.586	0.338	-2.887	0.024	-4.233	2	0.052

Table D.4: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *FH-compound* networks: The VBN4-8 model.

Appendix E

FH-compound networks: results by *HIDDEN* condition

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
full	0.844	0.838	0.812	2.8	1.6	2.7
Wd	0.844	0.838	0.812	2.8	1.6	2.7
Rnd	0.844	0.838	0.812	2.8	1.6	2.7
Rnd Wd	0.844	0.838	0.812	2.8	1.6	2.7
FH	0.791	0.821	0.726	3.3	1.7	3.2
FH Wd	0.781	0.822	0.750	3.4	1.7	3.1
FH Rnd	0.786	0.817	0.731	3.4	1.7	3.2
Cpos	0.752	0.752	0.509	4.1	2.2	7.4
Cpos Wd	0.738	0.737	0.512	4.3	2.3	8.8
Cpos Rnd	0.753	0.748	0.479	4.1	2.2	7.7
Cpos FH	0.802	0.824	0.757	3.2	1.7	3.1
Utt	0.844	0.838	0.812	2.8	1.6	2.7
Utt Wd	0.844	0.838	0.812	2.8	1.6	2.7
Utt Rnd	0.844	0.838	0.812	2.8	1.6	2.7
Utt FH	0.791	0.821	0.726	3.3	1.7	3.2
Utt Cpos	0.749	0.716	0.495	3.83	3.04	4.33
S	0.844	0.838	0.812	2.8	1.6	2.7
S Wd	0.844	0.838	0.812	2.8	1.6	2.7
S Rnd	0.844	0.838	0.812	2.8	1.6	2.7
S FH	0.791	0.821	0.726	3.3	1.7	3.2
S Cpos	0.739	0.726	0.494	4.4	2.4	6.4
S Utt	0.844	0.838	0.812	2.8	1.6	2.7
Wpos	0.844	0.838	0.812	2.8	1.6	2.7
Wpos Wd	0.844	0.838	0.812	2.8	1.6	2.7
Wpos Rnd	0.844	0.838	0.812	2.8	1.6	2.7
Wpos FH	0.738	0.822	0.752	4.0	1.7	3.1
Wpos Cpos	0.743	0.739	0.470	5.11	4.68	6.54
Wpos Utt	0.844	0.838	0.812	2.8	1.6	2.7
Wpos S	0.844	0.838	0.812	5.0	5.0	6.0

Table E.1: The correlation and RMS error results by voice by *HIDDEN. FH-compound* networks. The VBN1-8 model. *HIDDEN* observation condition.

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
full	0.843	0.836	0.812	2.8	1.6	2.7
Wd	0.850	0.836	0.812	2.8	1.6	2.7
Rnd	0.808	0.819	0.795	3.56	2.37	3.22
Rnd Wd	0.792	0.823	0.791	3.3	1.8	3.1
FH	0.814	0.823	0.776	3.0	1.7	3.0
FH Wd	0.753	0.819	0.757	3.6	1.7	3.1
FH Rnd	0.823	0.826	0.778	3.0	1.7	3.0
Cpos	0.770	0.737	0.627	3.9	2.3	4.5
Cpos Wd	0.764	0.736	0.615	4.1	2.3	4.5
Cpos Rnd	0.691	0.821	0.754	4.3	1.7	3.2
Cpos FH	0.795	0.821	0.721	3.3	1.7	3.4
Utt	0.850	0.836	0.812	2.8	1.6	2.7
Utt Wd	0.843	0.836	0.812	2.8	1.6	2.7
Utt Rnd	0.808	0.819	0.795	3.27	2.19	3.46
Utt FH	0.814	0.823	0.776	3.0	1.7	3.0
Utt Cpos	0.75	0.73	0.66	3.9	2.2	4.1
S	0.843	0.836	0.812	2.8	1.6	2.7
S Wd	0.843	0.836	0.812	2.8	1.6	2.7
S Rnd	0.804	0.823	0.793	3.18	2.17	3.45
S FH	0.808	0.826	0.773	3.1	1.7	3.2
S Cpos	0.774	0.762	0.620	3.8	2.1	4.4
S Utt	0.843	0.836	0.812	2.8	1.6	2.7
Wpos	0.843	0.836	0.812	2.8	1.6	2.7
Wpos Wd	0.843	0.836	0.812	2.8	1.6	2.7
Wpos Rnd	0.809	0.825	0.791	3.2	1.8	3.1
Wpos FH	0.747	0.820	0.783	3.7	1.7	3.0
Wpos Cpos	0.74	0.74	0.67	4.1	2.3	4.0
Wpos Utt	0.843	0.836	0.812	2.8	1.6	2.7
Wpos S	0.843	0.836	0.812	2.8	1.6	2.7

Table E.2: The correlation and RMS error results by voice by *HIDDEN. FH-compound* networks. The VBN2-8 model. *HIDDEN* observation condition.

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
full	0.828	0.816	0.756	4.0	2.5	4.2
Wd	0.828	0.816	0.756	4.0	2.5	4.2
Rnd	0.814	0.804	0.718	4.3	2.6	4.6
Rnd Wd	0.812	0.804	0.717	4.3	2.6	4.6
FH	0.828	0.816	0.756	4.0	2.5	4.2
FH Wd	0.828	0.816	0.756	4.0	2.5	4.2
FH Rnd	0.813	0.806	0.713	4.3	2.6	4.7
Cpos	0.776	0.716	0.632	5.4	3.7	5.3
Cpos Wd	0.773	0.712	0.628	5.5	3.7	5.5
Cpos Rnd	0.75	0.720	0.632	5.9	3.6	5.5
Cpos FH	0.776	0.725	0.630	5.5	3.5	5.5
Utt	0.828	0.816	0.756	4.0	2.5	4.2
Utt Wd	0.828	0.816	0.756	4.0	2.5	4.2
Utt Rnd	0.814	0.804	0.718	4.3	2.6	4.6
Utt FH	0.828	0.816	0.756	4.0	2.5	4.2
Utt Cpos	0.795	0.737	0.658	4.8	3.3	5.0
S	0.794	0.780	0.670	4.6	2.8	4.9
S Wd	0.788	0.775	0.673	4.7	2.9	4.8
S Rnd	0.768	0.764	0.674	5.0	3.0	5.0
S FH	0.789	0.775	0.669	4.7	2.9	4.9
S Cpos	0.762	0.709	0.609	5.8	3.9	5.4
S Utt	0.802	0.787	0.674	4.3	2.7	4.8
Wpos	0.805	0.786	0.660	4.8	2.9	5.1
Wpos Wd	0.806	0.799	0.654	4.8	2.8	5.1
Wpos Rnd	0.791	0.773	0.655	5.2	3.1	5.4
Wpos FH	0.805	0.799	0.657	4.8	2.8	5.1
Wpos Cpos	0.770	0.724	0.622	5.9	3.8	5.6
Wpos Utt	0.818	0.797	0.663	4.3	2.7	5.0
Wpos S	0.783	0.775	0.619	5.1	3.1	5.3

Table E.3: The correlation and RMS error results by voice by *HIDDEN. FH-compound* networks. The VBN3-8 model. *HIDDEN* observation condition.

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
full	0.842	0.843	0.796	4.1	2.4	4.7
Wd	0.785	0.806	0.771	4.8	2.7	4.9
Rnd	0.803	0.828	0.724	4.6	2.6	5.4
Rnd Wd	0.774	0.822	0.694	5.0	2.6	5.7
FH	0.768	0.822	0.575	5.2	2.5	6.9
FH Wd	0.745	0.818	0.535	5.6	2.6	7.3
FH Rnd	0.739	0.821	0.526	5.8	2.6	7.5
Cpos	0.735	0.768	0.529	5.6	3.0	7.5
Cpos Wd	0.739	0.767	0.486	5.5	2.9	7.9
Cpos Rnd	0.703	0.804	0.524	6.2	2.7	5.0
Cpos FH	0.754	0.812	0.551	5.5	2.6	9.1
Utt	0.779	0.783	0.702	5.3	3.0	5.6
Utt Wd	0.750	0.774	0.673	5.7	3.0	5.8
Utt Rnd	0.762	0.795	0.525	5.5	2.8	7.5
Utt FH	0.725	0.808	0.439	6.4	2.6	8.6
Utt Cpos	0.727	0.808	0.490	6.1	2.6	9.2
S	0.764	0.777	0.720	5.2	2.9	5.4
S Wd	0.751	0.754	0.674	5.62	3.31	6.01
S Rnd	0.756	0.805	0.649	5.3	2.7	6.2
S FH	0.721	0.819	0.503	6.0	2.6	7.5
S Cpos	0.698	0.756	0.449	6.2	3.0	8.1
S Utt	0.757	0.791	0.625	5.6	2.8	6.4
Wpos	0.756	0.760	0.652	5.4	3.1	6.2
Wpos Wd	0.749	0.766	0.590	5.4	2.9	6.9
Wpos Rnd	0.729	0.809	0.574	5.7	2.7	7.2
Wpos FH	0.717	0.816	0.555	6.3	2.6	7.0
Wpos Cpos	0.733	0.777	0.477	5.8	2.9	8.0
Wpos Utt	0.735	0.798	0.553	5.8	2.7	7.2
Wpos S	0.709	0.722	0.673	6.1	3.4	5.9

Table E.4: The correlation and RMS error results by voice by *HIDDEN. FH-compound* networks. The VBN4-8 model. *HIDDEN* observation condition.

Appendix F

MV-compound networks learnt by the K2 algorithm

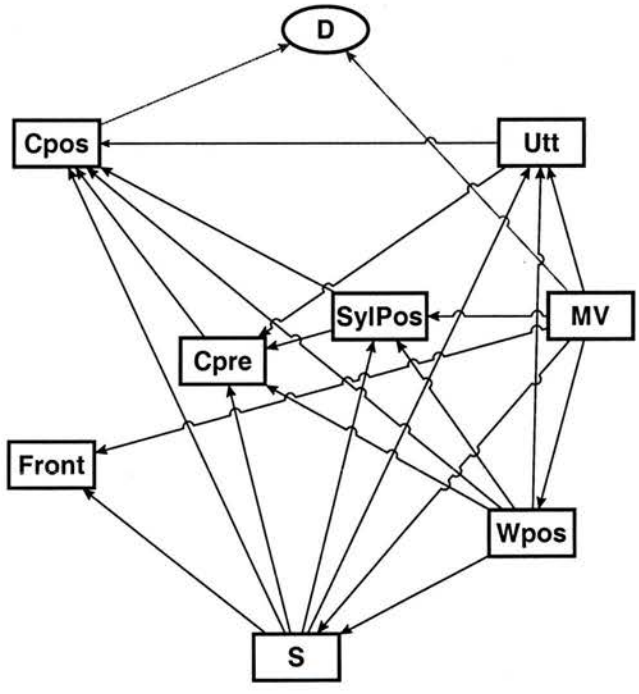

Figure F.1: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN1 model.

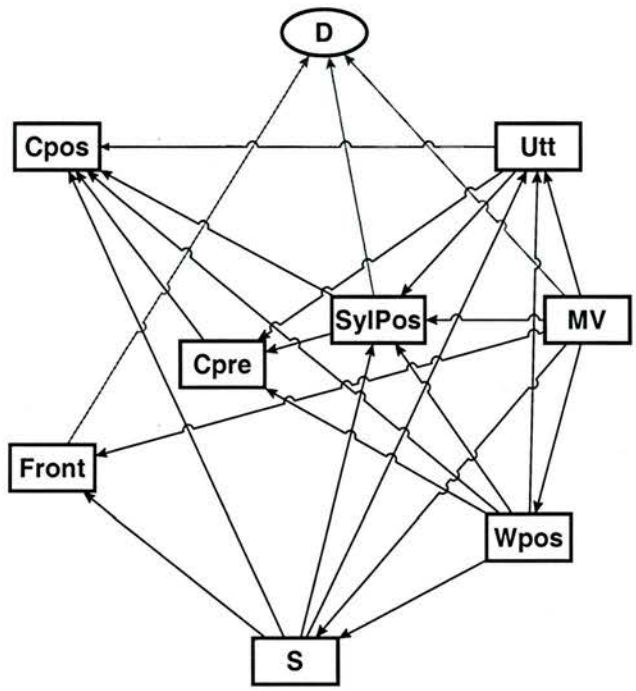

Figure F.2: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN2 model.

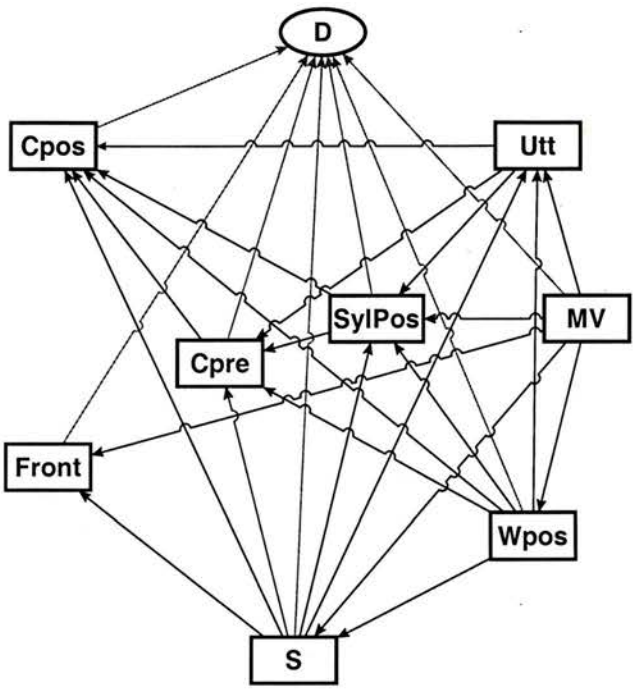

Figure F.3: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN3 model.

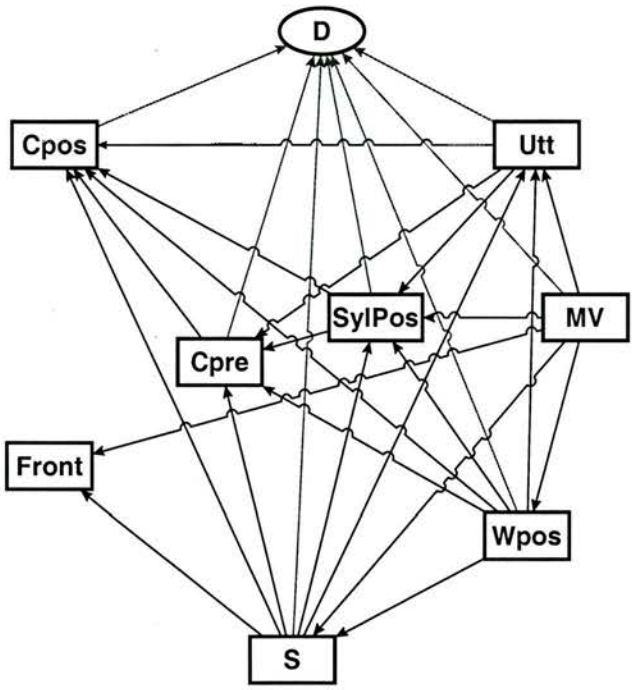

Figure F.4: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN4 model.

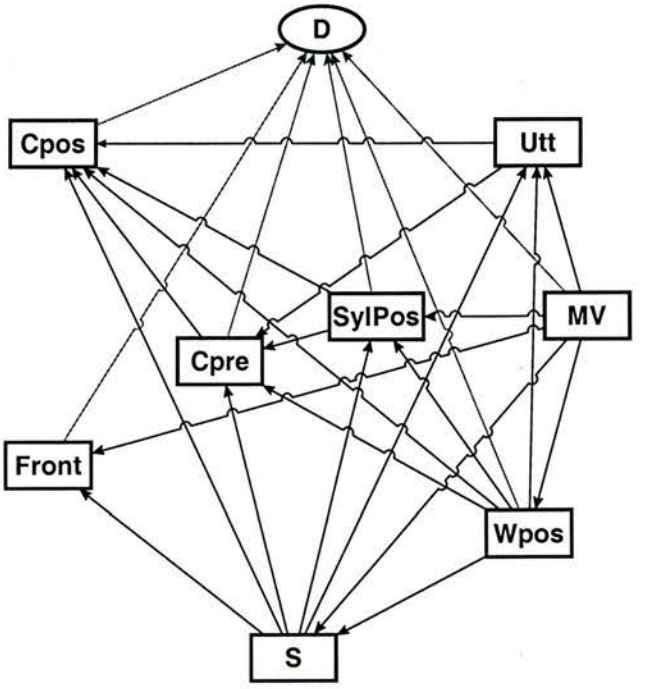

Figure F.5: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN6 model.

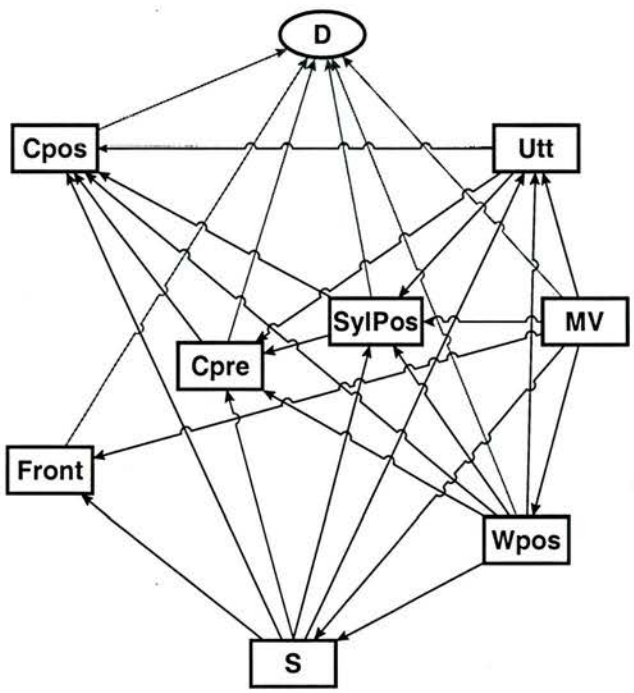

Figure F.6: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN7 model.

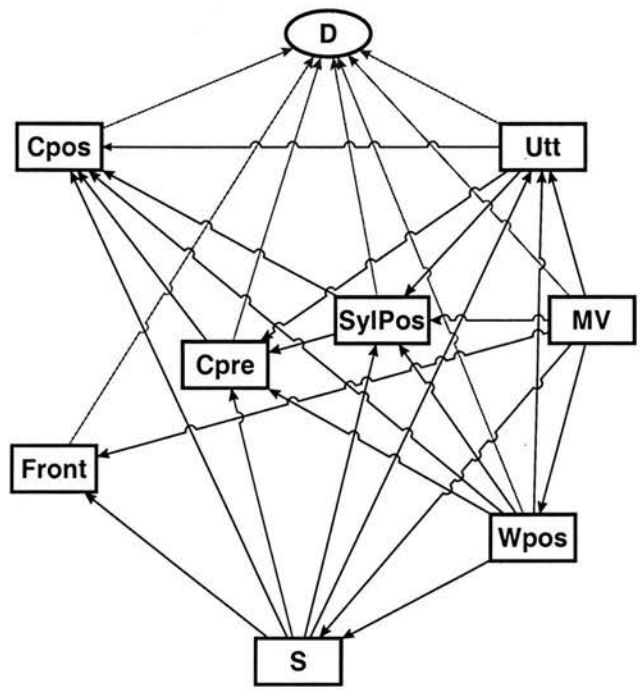

Figure F.7: *MV-compound* model learned by the K2 algorithm, with consonant durations being uniformly discretised. *MV-compound* CBN8 model.

Appendix G

MV-compound networks: results by *HIDDEN* condition

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
CBN1						
full	0.797	0.767	0.688	3.8	4.4	3.8
Wpos	0.767	0.717	0.634	3.7	4.3	3.8
Syl	0.797	0.767	0.688	4.4	5.0	4.5
Cpre	0.797	0.767	0.688	4.1	4.6	4.1
Cpos	0.762	0.723	0.634	4.2	4.8	4.2
Syl Cpre	0.797	0.767	0.688	5.0	5.6	5.0
Syl Cpos	0.797	0.767	0.688	5.5	6.1	5.6
CBN2						
full	0.727	0.759	0.665	5.1	5.6	5.1
Wpos	0.727	0.759	0.682	3.6	4.1	3.6
Syl	0.727	0.759	0.682	3.4	4.0	3.5
Cpre	0.703	0.711	0.603	4.3	4.8	4.3
Cpos	0.713	0.701	0.611	4.1	4.7	4.1
Syl Cpre	0.708	0.734	0.619	3.7	4.2	3.7
Syl Cpos	0.727	0.759	0.682	3.4	4.0	3.5
CBN3						
full	0.840	0.795	0.685	3.5	4.1	3.6
Wpos	0.741	0.689	0.679	3.6	4.2	3.7
Syl	0.773	0.715	0.683	3.5	4.1	3.6
Cpre	0.743	0.679	0.688	4.7	5.3	4.8
Cpos	0.692	0.647	0.532	4.8	5.4	5.0
Syl Cpre	0.724	0.622	0.630	4.5	5.2	4.6
Syl Cpos	0.791	0.736	0.747	4.4	5.0	4.4
CBN4						
full	0.724	0.744	0.803	4.6	5.1	4.5
Wpos	0.622	0.658	0.676	4.1	4.6	4.1
Syl	0.625	0.641	0.696	3.8	4.3	3.8
Cpre	0.561	0.555	0.627	5.0	5.6	5.0
Cpos	0.518	0.513	0.543	4.8	5.4	5.0
Syl Cpre	0.490	0.505	0.589	4.4	5.0	4.4
Syl Cpos	0.614	0.599	0.668	4.0	4.6	4.0

Table G.1: The correlation and RMS error results by voice. The *MV-compound* models. *HIDDEN* observation condition. Part 1.

Hidden	Correlation			RMSE		
	lja	rjs	erm	lja	rjs	erm
CBN5						
full	0.709	0.725	0.742	3.7	4.3	4.5
Wpos	0.579	0.636	0.620	4.2	4.7	4.1
Syl	0.623	0.623	0.625	4.3	4.9	4.2
Cpre	0.525	0.510	0.551	5.1	5.7	5.1
Cpos	0.477	0.487	0.430	4.2	4.8	4.2
Syl Cpre	0.458	0.501	0.541	4.9	5.6	4.9
Syl Cpos	0.532	0.562	0.606	4.1	4.7	4.0
CBN6						
full	0.804	0.740	0.752	4.6	5.2	4.6
Wpos	0.680	0.607	0.593	3.6	4.2	3.7
Syl	0.724	0.640	0.626	4.1	4.7	4.2
Cpre	0.685	0.600	0.627	3.8	4.4	3.9
Cpos	0.640	0.514	0.426	3.4	4.1	3.7
Syl Cpre	0.698	0.526	0.544	3.6	4.3	3.8
Syl Cpos	0.742	0.662	0.682	4.0	4.6	4.0
CBN7						
full	0.762	0.726	0.728	4.7	5.3	4.7
Wpos	0.620	0.613	0.595	5.6	6.2	5.7
Syl	0.737	0.681	0.648	4.7	5.4	4.8
Cpre	0.720	0.626	0.635	4.0	4.7	4.1
Cpos	0.675	0.564	0.440	4.7	5.4	5.0
Syl Cpre	0.718	0.614	0.537	3.9	4.6	4.1
Syl Cpos	0.720	0.675	0.661	5.1	5.7	5.1
CBN8						
full	0.561	0.494	0.750	3.5	4.1	3.4
Wpos	0.700	0.592	0.587	3.5	4.1	3.5
Syl	0.586	0.563	0.601	5.1	5.7	5.0
Cpre	0.571	0.498	0.594	4.2	4.8	4.1
Cpos	0.499	0.428	0.391	3.7	4.3	4.0
Syl Cpre	0.479	0.415	0.528	4.4	5.1	4.4
Syl Cpos	0.653	0.623	0.682	4.1	4.7	4.0

Table G.2: The correlation and RMS error results by voice. The *MV-compound* models. *HIDDEN* observation condition. Part 2.

Appendix H

MV-compound networks: correlation results

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.044	0.013	0.007	0.012	0.076	5.944	2	0.027
4	Cpos	0.044	0.009	0.005	0.021	0.067	8.277	2	0.014

Table H.1: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN1 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423
2	SylPos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423
3	Cpre	0.045	0.020	0.011	-0.004	0.093	3.930	2	0.059
4	Cpos	0.042	0.024	0.014	-0.018	0.102	2.990	2	0.096
5	Syl Cpre	0.030	0.015	0.008	-0.006	0.066	3.562	2	0.071
6	Syl Cpos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423

Table H.2: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN2 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.070	0.056	0.032	-0.069	0.210	2.174	2	0.162
2	SylPos	0.050	0.042	0.024	-0.054	0.154	2.058	2	0.176
3	Cpre	0.070	0.064	0.037	-0.088	0.228	1.903	2	0.197
4	Cpos	0.149	0.003	0.002	0.142	0.157	83.834	2	0.0001
5	SylPos Cpre	0.115	0.059	0.034	-0.031	0.260	3.393	2	0.077
6	SylPos Cpos	0.015	0.067	0.039	-0.151	0.182	0.398	2	0.729

Table H.3: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN3 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.114	0.025	0.014	0.052	0.176	7.910	2	0.016
2	SylPos	0.098	0.012	0.007	0.070	0.127	14.749	2	0.005
3	Cpre	0.183	0.007	0.004	0.166	0.199	47.255	2	0.0001
4	Cpos	0.241	0.017	0.010	0.199	0.282	25.132	2	0.002
5	Syl Cpre	0.234	0.019	0.011	0.187	0.282	21.113	2	0.002
6	Syl Cpos	0.152	0.022	0.013	0.098	0.206	12.162	2	0.007

Table H.4: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN4 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.105	0.016	0.010	0.064	0.146	11.015	2	0.008
2	SylPos	0.106	0.009	0.005	0.083	0.129	20.072	2	0.002
3	Cpre	0.190	0.026	0.015	0.126	0.254	12.762	2	0.006
4	Cpos	0.252	0.054	0.031	0.118	0.386	8.087	2	0.015
5	Syl Cpre	0.220	0.017	0.010	0.177	0.262	22.277	2	0.002
6	Syl Cpos	0.136	0.026	0.015	0.070	0.202	8.907	2	0.012

Table H.5: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN5 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.145	0.013	0.008	0.112	0.178	18.984	2	0.003
2	SylPos	0.084	0.053	0.030	-0.047	0.215	2.770	2	0.109
3	Cpre	0.102	0.053	0.031	-0.029	0.234	3.352	2	0.079
4	Cpos	0.213	0.120	0.069	-0.086	0.512	3.069	2	0.092
5	Syl Cpre	0.156	0.097	0.056	-0.084	0.395	2.793	2	0.108
6	Syl Cpos	0.063	0.019	0.011	0.016	0.110	5.798	2	0.028

Table H.6: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks: The CBN6 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.123	0.010	0.006	0.098	0.148	21.278	2	0.002
2	SylPos	0.068	0.020	0.011	0.019	0.117	5.951	2	0.027
3	Cpre	0.103	0.013	0.008	0.070	0.137	13.443	2	0.005
4	Cpos	0.204	0.072	0.041	0.026	0.383	4.926	2	0.039
5	Syl Cpre	0.136	0.047	0.027	0.019	0.254	4.995	2	0.038
6	Syl Cpos	0.060	0.008	0.005	0.040	0.080	12.787	2	0.006

Table H.7: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN7 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	-0.025	0.164	0.095	-0.432	0.382	-0.262	2	0.818
2	SylPos	0.018	0.115	0.066	-0.267	0.304	0.277	2	0.808
3	Cpre	0.047	0.094	0.054	-0.186	0.281	0.872	2	0.475
4	Cpos	0.162	0.170	0.098	-0.261	0.585	1.651	2	0.240
5	Syl Cpre	0.128	0.082	0.047	-0.076	0.331	2.698	2	0.114
6	Syl Cpos	-0.051	0.104	0.060	-0.310	0.209	-0.840	2	0.489

Table H.8: Paired (*FULL* vs. *HIDDEN*) t-test results for the correlation values. The *MV-compound* networks. The CBN8 model.

Appendix I

MV-compound networks: RMS error results

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	-0.327	0.130	0.075	-0.650	-0.005	-4.370	2	0.049
4	Cpos	-0.297	0.124	0.072	-0.605	0.011	-4.150	2	0.053

Table I.1: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN1 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423
2	SylPos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423
3	Cpre	0.045	0.020	0.011	-0.004	0.093	3.930	2	0.059
4	Cpos	0.042	0.024	0.014	-0.018	0.102	2.990	2	0.096
5	SylPosCpre	0.030	0.015	0.008	-0.006	0.066	3.562	2	0.071
6	SylPosCpos	-0.006	0.010	0.006	-0.029	0.018	-1.000	2	0.423

Table I.2: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN2 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.070	0.056	0.032	-0.069	0.210	2.174	2	0.162
2	SylPos	0.050	0.042	0.024	-0.054	0.154	2.058	2	0.176
3	Cpre	0.070	0.064	0.037	-0.088	0.228	1.903	2	0.197
4	Cpos	0.149	0.003	0.002	0.142	0.157	83.834	2	0.000
5	SylPosCpre	0.115	0.059	0.034	-0.031	0.260	3.393	2	0.077
6	SylPosCpos	0.015	0.067	0.039	-0.151	0.182	0.398	2	0.729

Table I.3: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN3 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.114	0.025	0.014	0.052	0.176	7.910	2	0.016
2	SylPos	0.098	0.012	0.007	0.070	0.127	14.749	2	0.005
3	Cpre	0.183	0.007	0.004	0.166	0.199	47.255	2	0.000
4	Cpos	0.241	0.017	0.010	0.199	0.282	25.132	2	0.002
5	SylPosCpre	0.234	0.019	0.011	0.187	0.282	21.113	2	0.002
6	SylPosCpos	0.152	0.022	0.013	0.098	0.206	12.162	2	0.007

Table I.4: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN4 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.105	0.016	0.010	0.064	0.146	11.015	2	0.008
2	SylPos	0.106	0.009	0.005	0.083	0.129	20.072	2	0.002
3	Cpre	0.190	0.026	0.015	0.126	0.254	12.762	2	0.006
4	Cpos	0.252	0.054	0.031	0.118	0.386	8.087	2	0.015
5	SylPosCpre	0.220	0.017	0.010	0.177	0.262	22.277	2	0.002
6	SylPosCpos	0.136	0.026	0.015	0.070	0.202	8.907	2	0.012

Table I.5: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN5 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.145	0.013	0.008	0.112	0.178	18.984	2	0.003
2	SylPos	0.084	0.053	0.030	-0.047	0.215	2.770	2	0.109
3	Cpre	0.102	0.053	0.031	-0.029	0.234	3.352	2	0.079
4	Cpos	0.213	0.120	0.069	-0.086	0.512	3.069	2	0.092
5	SylPosCpre	0.156	0.097	0.056	-0.084	0.395	2.793	2	0.108
6	SylPosCpos	0.063	0.019	0.011	0.016	0.110	5.798	2	0.028

Table I.6: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN6 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	0.123	0.010	0.006	0.098	0.148	21.278	2	0.002
2	SylPos	0.068	0.020	0.011	0.019	0.117	5.951	2	0.027
3	Cpre	0.103	0.013	0.008	0.070	0.137	13.443	2	0.005
4	Cpos	0.204	0.072	0.041	0.026	0.383	4.926	2	0.039
5	SylPosCpre	0.136	0.047	0.027	0.019	0.254	4.995	2	0.038
6	SylPosCpos	0.060	0.008	0.005	0.040	0.080	12.787	2	0.006

Table I.7: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN7 model.

#	Hidden	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
1	Wpos	-0.025	0.164	0.095	-0.432	0.382	-0.262	2	0.818
2	SylPos	0.018	0.115	0.066	-0.267	0.304	0.277	2	0.808
3	Cpre	0.047	0.094	0.054	-0.186	0.281	0.872	2	0.475
4	Cpos	0.162	0.170	0.098	-0.261	0.585	1.651	2	0.240
5	SylPosCpre	0.128	0.082	0.047	-0.076	0.331	2.698	2	0.114
6	SylPosCpos	-0.051	0.104	0.060	-0.310	0.209	-0.840	2	0.489

Table I.8: Paired (*FULL* vs. *HIDDEN*) t-test results for the RMS error values. The *MV-compound* networks. The CBN8 model.

References

Allen, J., Hunnicut, S. & Klatt, D. (1987), *From Text to Speech: The MITalk System*, Cambridge University Press, Cambridge.

Bagshaw, P. (1998), Unsupervised training of phone duration and energy models for text-to-speech synthesis, *in* 'Proceedings of the 5th International Conference on Spoken Language Processing', Vol. 2, Sydney, Australia, pp. 132–135.

Barbosa, P. & Bailly, G. (1994), 'Characterisation of rhythmic patterns for text-to-speech synthesis', *Speech Communication* **15**, 127–137.

Bell, A., Jurafsky, D., Fosler-Lussier, E., Girand, C., Gregory, M. & Gildea, D. (2003), 'Effects of disfluencies, predictability, and utterance position on word form variation in English conversation', *The Journal of the Acoustic Society of America* **113**(2), 1001–1024.

Beutnagel, M., Mohri, M. & Riley, M. (1999), Rapid unit selection from large corpus for concatenative speech synthesis, *in* 'Proceedings Eurospeech 99', Vol. 2, Budapest, Hungary, p. 607.

Bishop, C. (1998), *Neural Networks for Pattern Recognition*, Clarendon Press, Cambridge.

Black, A., Caley, R., King, S. & Taylor, P. (2003), Edinburgh Speech Tools Library: system documentation, Technical Report 1.2.0 edition, The Centre for Speech Technology Research, University of Edinburgh, UK.

Black, A., Taylor, P. & Caley, R. (2000), The Festival speech synthesis system: system documentation, Technical Report 1.4.0 edition, The Centre for Speech Technology Research, University of Edinburgh, UK.

Bouckaert, R. (1994), Probabilistic network construction using the minimum description length principle, Technical Report RUU-CS-94-27, Utrecht University, Netherlands.

Boutilier, C., Friedman, N., Goldszmidt, M. & Koller, D. (1996), Context specific independence in baysian networks, *in* 'Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI)', Portland, Oregon, USA, San Francisco, CA, pp. 115–123.

Breiman, L., Friedman, J. & Olshen, R. (1984), *Classification and Regression Trees*, Wadsworth and Brooks, Pacific Grove.

Campbell, W. & Isard, S. (1991), 'Segment durations in a syllable frame', *Journal of Phonetics* **19**, 37–47.

Chickering, D. (2002), 'Learning equivalence classes of Bayesian-network structures', *Machine Learning* **2**, 445–498.

Coker, C., Umeda, N. & Browman, C. (1973), 'Automatic synthesis from ordinary English text', *IEEE Transactions on Audio and Electroacoustics* **AU-21**, 293–298.

Coombs, C. (1964), *A theory of data*, Wiley, New York.

Cooper, A. (1991), Laryngeal and oral gestures in englush /p, t, k/, *in* 'Proceedings of the XIIth International Congress of Phonetic Sciences', Vol. 2, Aix-en-Provance, France, pp. 50–53.

Cooper, G. & Herskovits, E. (1992), 'A bayesian method for the induction of probabilistic networks from data', *Machine Learning* **9**, 309–347.

Cowell, R., Dawid, A., Lauritzen, S. & Spiegelhalter, D. (1999), *Probabilistic networks and expert systems*, Springer, New York.

Crystal, T. & House, A. (1988*a*), 'Segmental durations in connected-speech signals: Syllabic stress', *The Journal of the Acoustical Society of America* **83**(4), 1574–1585.

Crystal, T. & House, A. (1988*b*), 'Segmental durations in connected-speech sygnals: Current results', *The Journal of the Acoustical Society of America* **83**(4), 1553–1573.

Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete date via the em algorithm', *Journal of the Royal Statistical Society* **B 39**, 1–38.

Fougeron, C. & Keating, P. (1997), 'Articulatory strengthening at edges of prosodic domains', *The Journal of the Acoustical Society of America* **101**(6), 3728–3740.

Friedman, N. & Goldszmidt, M. (1996), Learning Baysian networks with local structure, *in* 'Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI)', Morgan Kaufmann Publishers, San Francisco, CA, pp. 252–262.

Goubanova, O. (2001), Predicting segmental duration using Bayesian belief networks, *in* 'Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis', Perthshire, Scotland, pp. 139–142.

Goubanova, O. (2003), Bayesian modelling of vowel segment duration for text-to-speech synthesis using distinctive features, *in* 'Proceedings of 15th International Congress of Phonetic Sciences', Vol. 4, Barcelona, Spain, pp. 1941–1944.

Goubanova, O. & King, S. (2005), Predicting consonant duration with Bayesian belief networks, *in* 'Proceedings of the Interspeech 2005', Vol. 4, Lisbon, Portugal, pp. 1941–1944.

Goubanova, O. & Taylor, P. (2000), Using Bayesian belief networks for model duration in text-to-speech systems, *in* 'Proceedings of the 6th International Conference on Spoken Language Processing', Vol. 4, Beijing, China, pp. 1126–1129.

Gregory, M., Bell, A., Jurafsky, D. & Raymond, W. (2001), 'Frequency and predictability effects on the duration of content words in conversation', *The Journal of the Acoustic Society of America* **110**(5), 2738.

Haggard, D. (1973), 'Abbreviation of consonants in English pre- and post-vocalic clusters', *Journal of Phonetics* **1**(1), 9–24.

Hamza, W. & Donovan, R. (2002), Data-driven segemnt preselection in the IBM trainable speech synthesis system, *in* 'Proceedings of the 7th International Conference on Spoken Language Processing', Vol. 6, Denver, Colorado, USA, pp. 2609–2612.

Heckerman, D. (1995), A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95-06, Microsoft Reserach, Microsoft Corporation, Redmond, USA.

Hofer, G., Richmond, K. & Clark, R. (2005), Informed blending of databases for emotional speech synthesis, *in* 'Proceedings of Interspeech 2005', Vol. 1, Lisbon, Portugal, pp. 501–504.

Huang, C. & Darwiche, A. (1996), 'Inference in belief networks: A procedural guide', *International Journal of Approximate Reasoning* **15**(3), 225–263.

Jensen, F. (1996), *Introduction to Bayesian Networks*, UCL Press, London.

Jordan, M., ed. (1999), *Learning in Graphical Models*, The MIT Press, Cambridge.

Kaiki, N., Takeda, K. & Sagisaka, Y. (1990), Statistical analysis for segmental duration rules in japanese, *in* 'Proceedings of the 1st International Conference on Spoken Language Processing', Kobe, Japan, pp. 17–20.

Kjaerulff, U. (1992), 'Optimal decomposition of probability networks by simulated annealing', *Statistics and Computing* **2**, 7–17.

Klabbers, E. & Veldhui (1996), On the reduction of concatenation artefacts in diphone synthesis, *in* 'Proceedings of International Conference on Spoken Language Processing', Vol. 6, Sydney, Australia, pp. 1983–1986.

Klatt, D. (1973), 'Interaction between two factors that influence vowel duration', *The Journal of the Acoustic Society of America* **54**(4), 1102–1104.

Klatt, D. (1974), 'The duration of [s] in English words', *Journal of Speech and Hearing Research* **17**, 51–63.

Klatt, D. (1975), 'Vowel lengthening is syntactically determined in connnected speech', *Journal of Phonetics* **59**(3), 129–140.

Klatt, D. (1976), 'Linguistic uses of segmental duration of English: Acoustic and perceptual evidence', *The Journal of the Acoustic Society of America* **59**(5), 1209–1211.

Krantz, D., Luce, R., Suppes, P. & Tversky, A. (1964), *Foundations of measurement*, Vol. 1, Wiley, New York.

Lam, W. & Bachus, F. (1994), 'Learning Bayesian belief networks: An approach based on the MDL principle', *Computational Intelligence* **10**, 269–293.

Lauritzen, S. & Spiegelhalter, D. (1988), 'Local computations with probabilities on graphical structures and their application to expert systems', *Journal of the Royal Statistical Society* **50**(B), 157–224.

Lee, P. M. (1997), *Bayesian Satistics*, Arnold, Cambridge.

Lehiste, I. (1972), 'The timing of utterances and linguistic boundaries', *The Journal of the Acoustical Society of America* **51**(6B), 2018–2024.

Lehiste, I. (1973), 'Rhythmic units and syntactic units in production and perception', *The Journal of the Acoustical Society of America* **54**(5), 1228–1234.

Lindblom, D. & Rapp, K. (1973), Some temporal regularities of spoken Swedish, *in* 'PILUS', Vol. 21, Sweden, pp. 1–59.

Mayo, C., Clark, R. & King, S. (2005), Multidimensional scaling of listener responses to synthetic speech, *in* 'Proceedings Interspeech 2005', Vol. 4, Lisbon, Portugal, pp. 1725–1728.

Nooteboom, S. (1972), Production and perception of vowel duration, PhD thesis, University of Utrecht.

Olesen, K. (1993), 'Causal probabilistic networks with discrete and continuous variables', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(3), 275–279.

Oller, O. (1973), 'The effect of position in utterance on speech segment duration in English', *The Journal of the Acoustical Society of America* **54**(5), 1235–1247.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann.

Peterson, G. & Lehiste, I. (1960), 'Duration of syllable nuclei in English', *The Journal of the Acoustical Society of America* **32**, 693–703.

Port, R. (1981), 'Linguistic timing factors in combination', *The Journal of the Acoustical Society of America* **69**(1), 262–273.

Raux, A. & Black, A. (2003), A unit selection approach to F0 modelling and its application to emphasis, *in* 'Proceedings of 2003 IEEE Workshop on Automatic Speech Understanding and Recognition', St. Thomas, Virgin Isles, USA.

Shih, C. & van Santen, J. (2000), 'Suprasegmental and segmental timing models in Mandarin Chinese and American English', *The Journal of the Acoustical Society of America* **107**(2), 1012–1026.

Sluijter, A. & van Heuven, V. (1995), 'Effects of focus distribution, pitch accent and lexical stress on the temporal organization of syllables in dutch', *Phonetica* **52**, 71–89.

Stylianou, Y. & Syrdal, A. (2001), Perceptual and objective detection of discontinuities in concatenative speech synthesis, *in* 'Proceedings of ICASSP', Vol. 2, Salt Lake City, USA, pp. 2090–2093.

Turk, A. & Shattuck-Hufnagel, S. (2000), 'Word-boundary-related duration patterns in English', *Journal of Phonetics* **28**(4), 397–440.

Turk, A. & White, L. (1999), 'Structural influenceson accentual lengthening in English', *Journal of Phonetics* **27**(2), 171–206.

Umeda, N. (1975*a*), 'Another consistency in phoneme duration', *The Journal of the Acoustical Society of America* **58**(S1), 62.

Umeda, N. (1975*b*), 'Vowel duration in American English', *The Journal of the Acoustical Society of America* **58**(2), 435–445.

Umeda, N. (1977), 'Consonant duration in American English', *The Journal of the Acoustical Society of America* **61**(3), 847–858.

van Santen, J. (1992*a*), Diagnostic perceptual experiments for text-to-speech system evaluation, *in* 'Proceedings of the 2nd International Conference on Spoken Language Processing', Vol. 3, Banff, Canada, pp. 555–558.

van Santen, J. H. (1994), 'Assignment of segmental duration in text-to-speech synthesis', *Computer Speech and Language* **8**, 95–128.

van Santen, J. P. H. (1992*b*), 'Contextual effects on vowel durations', *Speech Communication* **11**, 513–546.

van Santen, J. P. H. (1993), 'Analysing n-way tables with sums-of-products models', *Journal of Mathmatical Psychology* **37**, 327–371.

van Son, R. & van Santen, J. (1997), Strong interaction between factors influencing consonant duration, *in* 'Proceedings of the Interspeech'97', Rhodes, Greece, pp. 319–322.

Vepa, J. & King, S. (To appear 2005), Subjective evaluation of join cost and smoothing methods for unit selection speech synthesis, *in* 'IEEE Transactions on Speech and Audio Processing', Sydney, Australia.

Wightman, C., Shattuck-Hufnagel, S., Ostendorf, M. & Price, P. (1992), 'Segmental durations in the vicinity of procodic phrase boundaries', *The Journal of the Acoustical Society of America* **91(3)**, 1707–1717.